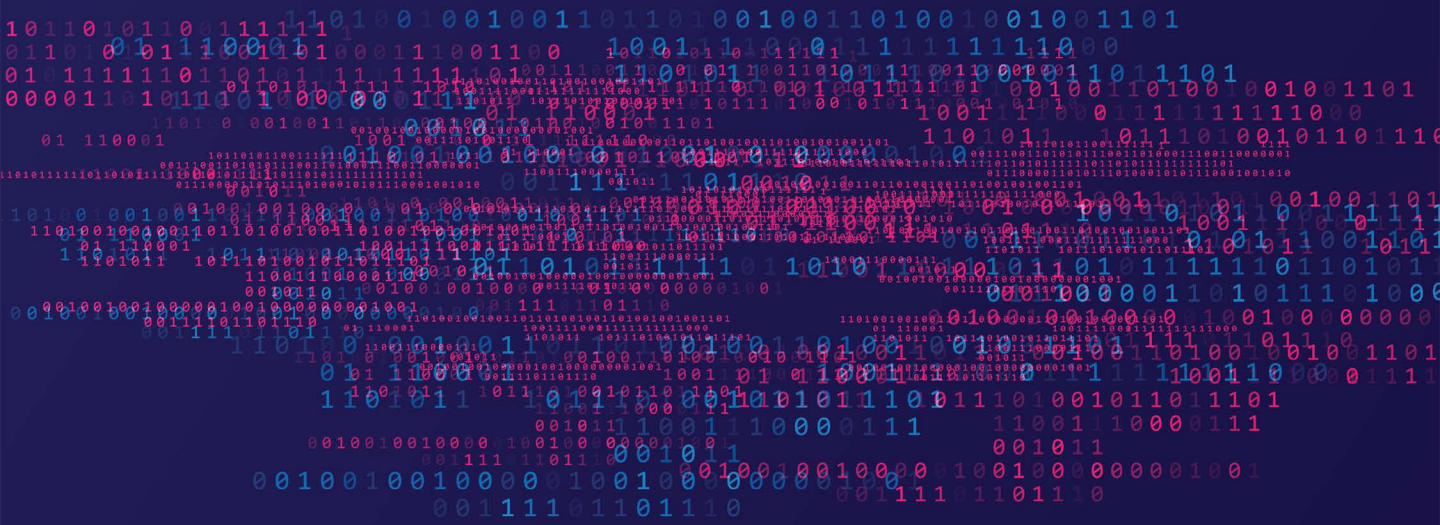
EXPERT INSIGHT

# The Software Developer's Guide to Linux

**A practical, no-nonsense guide to using the Linux command line and utilities as a software developer**



**David Cohen**
**Christian Sturm**

‹packt›

# The Software Developer's Guide to Linux

A practical, no-nonsense guide to using the Linux command line and utilities as a software developer

**David Cohen**

**Christian Sturm**

‹packt›

BIRMINGHAM—MUMBAI

# The Software Developer's Guide to Linux



**Senior Publishing Product Manager**: Aaron Tanna
**Acquisition Editor – Peer Reviews**: Gaurav Gavas
**Project Editor**: Parvathy Nair
**Content Development Editor**: Matthew Davies
**Copy Editor**: Safis Editing
**Technical Editor**: Karan Sonawane
**Proofreader**: Safis Editing
**Indexer**: Tejal Soni
**Presentation Designer**: Ganesh Bhadwalkar
**Developer Relations Marketing Executive**: Meghal Patel

First published: January 2024
Production reference: 1230124

Published by Packt Publishing Ltd.
Grosvenor House
11 St Paul's Square
Birmingham
B3 1RB, UK.

ISBN 978-1-80461-692-5

www.packt.com

# Contributors

## About the authors

**David Cohen** has, for the past 15 years, worked as a Linux system administrator, software engineer, infrastructure engineer, platform engineer, site reliability engineer, security engineer, web developer, and a few other things besides. In his free time, he runs the *tutorialinux* YouTube channel where he's taught hundreds of thousands of people the basics of Linux, programming, and DevOps. David has been at Hashicorp since 2019—first as an SRE, then as a reference architect, and now as a software engineer.

*Thank you, Aleyna, for your unwavering support over the past few years as I've been developing and writing this book. Without you, this would just be another of my promising-but-unfinished projects languishing in some forgotten "Archive" directory. Thanks to Christian, who has stuck with me for over a decade as a friend and a partner on practically every wild tech project idea I've come up with since we met. Finally, a big "thank you" is also due to my friends and colleagues at Hashicorp and everywhere else I've been over the past 15 years, who have made me a better engineer and encouraged projects like this.*

**Christian Sturm** is a consultant on software and systems architecture, having worked in various technical positions for well over a decade. He has worked as an application developer for the frontend and backend at companies large and small, such as zoomsquare and Plutonium Labs. On top of that, he is also an active contributor to various open source projects and has a deep understanding of fields including operating systems, networking protocols, security, and database management systems.

## About the reviewers

**Mario Splivalo** works as a consultant dealing with databases extended into modern cloud-based architectures. He also helps companies design their infrastructure using IaaC tools such as *Terraform* and *AWS Cloudformation*. For five years, Mario worked with *Canonical* as an OpenStack engineer.

Mario's fascination with computers started back when Commodore 64 dominated the user space. He took his first steps using BASIC on his dad's C64, quickly shifting to Assembler. He gradually moved to PCs, finding a great love for programming, systems design, and database administration. He switched to Linux (Knoppix, then Ubuntu, and never looked back) in the early 2000s, continuing as a database administrator, programmer, and system administrator.

**Nathan Chancellor** is an independent contractor working on the Linux kernel, based in Arizona, US. As a developer, his focus is on improving the compatibility between the Linux kernel and the LLVM toolchain. He has used Linux since 2016 and it has been his primary development operating system since 2018. His distributions of choice are Arch Linux and Fedora.

## Learn more on Discord

To join the Discord community for this book – where you can share feedback, ask questions to the author, and learn about new releases – follow the QR code below:

`https://packt.link/SecNet`

# Table of Contents

## Chapter 12: Automating Tasks with Shell Scripts 151

# Preface

Many software engineers are new to Unix-like systems, even though these systems are everywhere in the software engineering world. Whether developers know it or not, they're expected to work with Unix-like systems running in their work environment (macOS), their software development process (Docker containers), their build and automation tooling (CI and GitHub), their production environments (Linux servers and containers), and more.

Being skilled with the Linux command line can help software developers go beyond what's expected of them, allowing them to:

- Save time by knowing when to use built-in Unix tools, instead of writing thousand-line scripts or helper programs
- Help debug complex production outages, often involving Linux servers and their interface to the application
- Mentor junior engineers
- Have a more complete understanding of how the software they write fits into the larger ecosystem and tech stack

We hope that the theory, examples, and projects included in this book can take your Linux development skills to the next level.

## Who this book is for

This book is for software developers who are new to Linux and the command line, or who are out of practice and want to quickly dust off their skills. If you still feel a bit insecure about your abilities when you're staring at a Linux command-line prompt on a production server at 2:00 in the morning, this book is for you. If you want to quickly fill a Linux skills gap to advance your career, this book is for you. If you're just curious, and you want to see what kind of efficiency gains you can make in your day-to-day development setup and routines by adding some command-line magic, this book will serve you as well.

## What this book is not

One of the ways we have tried to fulfill our vision for this kind of uniquely useful book is by being extremely careful about what's included. We've tried to cut out everything that isn't essential to your life as a developer, or to a basic understanding of Linux and its core abstractions. In other words, the reason this book is useful is because of *all the things we left out*.

This book is not a full Linux course. It's not for people working as Linux system engineers or kernel developers. Because of this, it's not 750+ pages long, and you should be able to work through it in a few days, perhaps during a quiet sprint at work.

## What this book covers

*Chapter 1*, *How the Command Line Works*, explains how a command-line interface works, what a shell is, and then immediately gives you some basic Linux skills. You'll get a bit of theory and then begin moving around on the command line, finding and working with files and learning where to look for help when you get stuck. This chapter caters to new developers by teaching the most important command-line skills. If you read nothing else, you'll still be better off than when you started.

*Chapter 2*, *Working with Processes*, will take you on a guided tour of Linux processes. You'll then dive into useful, practical command-line skills for working with processes. We'll add detail to a few aspects that are a common source of process-related problems that you'll encounter as a software developer, like permissions, and give you some heuristics for troubleshooting them. You'll also get a quick tour of some advanced topics that will come up again later in the book.

*Chapter 3*, *Service Management with systemd*, builds on the knowledge about processes learned in the previous chapter by introducing an additional layer of abstraction, the `systemd` service. You'll learn about what an `init` system does for an operating system, and why you should care. Then, we cover all the practical commands you'll need for working with services on a Linux system.

*Chapter 4*, *Using Shell History*, is a short chapter covering some tricks that you can learn to improve your speed and efficiency on the command line. These tricks revolve around using shortcuts and leveraging shell history to avoid repeated keystrokes.

*Chapter 5*, *Introducing Files*, introduces files as the essential abstraction through which to understand Linux. You'll be introduced to the **Filesystem Hierarchy Standard** (**FHS**), which is like a map that you can use to orient yourself on any Unix system. Then it's time for practical commands for working with files and directories in Linux, including some special filetypes you probably haven't heard of. You'll also get a taste of searching for files and file content, which is one of the most powerful bits of knowledge to have at your fingertips as a developer.

*Chapter 6*, *Editing Files on the Command Line*, introduces two text editors – nano and vim. You will learn the basics of using these text editors for command-line editing while also becoming aware of common editing mistakes and how to avoid them.

*Chapter 7*, *Users and Groups*, will introduce you to how the concepts of users and groups form the basis for the Unix security model, controlling access for resources like files and processes. We'll then teach you the practical commands you'll need to create and modify users and groups.

*Chapter 8*, *Ownership and Permissions*, builds on the previous chapter's explanation of users and groups to show you how access control works for resources in Linux. This chapter teaches you about ownership and permissions by walking you through file information from a long listing. From there, we'll look at the common file and directory permissions that you'll encounter on production Linux systems, before engaging with the Linux commands for modifying file ownership and permissions.

*Chapter 9*, *Managing Installed Software*, shows you how to install software on various Linux distributions (and even macOS). First, we introduce package managers, which are the preferred way of getting software onto a machine: you'll learn the important theory and practical commands for the package management operations you'll need as a software developer. Then we'll introduce a few other methods, like downloading install scripts and the time-honored, artisanal Unix tradition of compiling your own software locally, from source (it's not as scary as it sounds!).

*Chapter 10*, *Configuring Software*, piggybacks off the previous chapter's focus on installing software by helping you with configuring software on a Linux system. You will learn about the places that most software will look for configuration ("the configuration hierarchy"). Not only will this knowledge come in handy during late-night troubleshooting sessions, but it can actually help you to write better software. We'll cover command-line arguments, environment variables, configuration files, and how all of this works on non-standard Linux environments like Docker containers. There's even a little bonus project: you'll see how to take a custom program and turn it into its own `systemd` service.

*Chapter 11*, *Pipes and Redirection*, will give you an introduction to what is possibly the "killer feature" of Unix: the ability to connect existing programs into a custom solution using pipes. We'll move through the prerequisite theory and practical skills you need to understand: file descriptors and input/output redirection. Then you'll jump into creating complex commands using pipes. You'll be introduced to some essential CLI tools and practical pipe patterns, which you'll still find yourself using long after you finish this book.

*Chapter 12*, *Automating Tasks with Shell Scripts*, serves as a Bash scripting crash course, teaching you how to go from typing individual commands in an interactive shell to writing scripts. We assume you're already a software developer, so this will be a quick introduction that shows you the core language features and doesn't spend a lot of time re-explaining the basics of programming. You'll learn about Bash syntax, best practices for script writing, and some important pitfalls to avoid.

*Chapter 13*, *Secure Remote Access with SSH*, explores the Secure Shell Protocol and the related command-line tools available to you. You'll learn the basics of **public-key cryptography** (**PKI**), which is always useful for a developer to know, before diving into creating SSH keys and securely logging into remote systems over the network. You'll build on this knowledge and get some experience copying files over the network, using SSH to create ad-hoc proxies or VPNs, and see examples of various other tasks that involve moving data over an encrypted SSH tunnel.

*Chapter 14*, *Version Control with Git*, shows you how to use a tool you probably already know well – `git` – from the command line, instead of through your IDE or a graphical client. We quickly go through the basic theory behind git and then jump into the commands you'll need to use in a command-line environment. We'll cover two powerful features that it pays to understand – bisecting and rebasing – and then give you our take on best practices and useful shell aliases. Finally, the *Poor man's GitHub* section presents a small but legitimately useful project that you can do to practice and integrate the Linux skills you've learned up to this point.

*Chapter 15*, *Containerizing Applications with Docker*, gives you the basic theory and practical skills that will make it easy to work with Docker as a developer. We'll explore the problems that Docker solves, explain the most important Docker concepts, and take you through the core workflow and commands you'll use. You'll also see how to build your own images by containerizing a real application. And because we're approaching this from a software development and Linux perspective, you'll also develop a good intuition for how containerization works under the hood, and how it's different from virtual machines.

*Chapter 16*, *Monitoring Application Logs*, gives an overview of logging on Unix and Linux. We'll show you how (and where) logs are collected on most modern Linux systems using `systemd`, and how more traditional approaches work (you'll come across both in the real world). You'll build practical command-line skills finding and viewing logs and learn a bit about how logging is being done in larger infrastructures.

*Chapter 17*, *Load Balancing and HTTP*, covers the basics of HTTP for developers, with a special focus on the complexities that you'll come across when working with HTTP services in larger infrastructures. We'll correct some common misunderstandings about HTTP statuses, HTTP headers, and HTTP versions and how applications should handle them. We'll also introduce how load balancers and proxies work in the real world, and how they make the experience of troubleshooting a live application quite different from troubleshooting a development version on your laptop. Many of the Linux skills that you will have learned up to this point will come in handy here, and we'll introduce a new tool – `curl` – to help you troubleshoot a wide variety of HTTP-related issues.

# To get the most out of this book

If you can get yourself to a Linux shell prompt – by installing Ubuntu in a virtual machine or running it as a Docker container, for example – you can follow along with everything in this book.

You can get away with even less – on Windows, there's WSL, and macOS is a bona-fide Unix operating system, so almost all of the practical commands you learn in this book (except those called out as Linux-only) will work out of the box. That said, for the best experience, follow along on a Linux operating system.

The skills required to get the most out of this book are only the basic computer skills that you already have as a software developer – editing text, working with files and folders, having some notion of what "operating systems" are, installing software, and using a development environment. Everything beyond that, we'll teach you.

## Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: `https://packt.link/gbp/9781804616925`.

## Conventions used

There are a number of text conventions used throughout this book.

`CodeInText`: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. For example: "The `-f` flag stands for 'follow,' and the `-u` flag stands for 'unit.'"

A block of command line is set as follows:

```
/home/steve/Desktop# ls
anotherfile  documents  somefile.txt  stuff
/home/steve/Desktop# cd documents/
/home/steve/Desktop/documents# ls
contract.txt
```

**Bold**: Indicates a new term, an important word, or words that you see on the screen For instance, words in menus or dialog boxes appear in the text like this. For example: "When a file is set to be executable, Unix will do its best to execute it, either succeeding in the case of **ELF** (**Executable and Linkable Format**, probably the most widely used executable format today) or failing."

Warnings or important notes appear like this.

Tips and tricks appear like this.

## Get in touch

Feedback from our readers is always welcome.

**General feedback**: Email `feedback@packtpub.com`, and mention the book's title in the subject of your message. If you have questions about any aspect of this book, please email us at `questions@packtpub.com`.

**Errata**: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book we would be grateful if you would report this to us. Please visit, `http://www.packtpub.com/submit-errata`, selecting your book, clicking on the Errata Submission Form link, and entering the details.

**Piracy**: If you come across any illegal copies of our works in any form on the Internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packtpub.com` with a link to the material.

**If you are interested in becoming an author**: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `http://authors.packtpub.com`.

## Share your thoughts

Once you've read *The Software Developer's Guide to Linux*, we'd love to hear your thoughts! Please `click here to go straight to the Amazon review page` for this book and share your feedback.

Your review is important to us and the tech community and will help us make sure we're delivering excellent quality content.