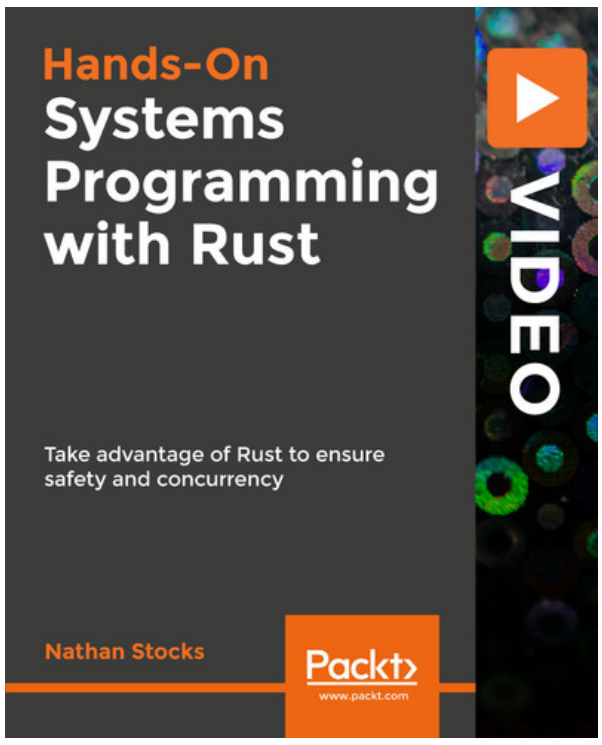
Hands-On Systems Programming with Rust

★★★★★ 17 reviews

By Nathan Stocks



TIME TO COMPLETE:
2h 15m

LEVEL:
Intermediate

TOPICS:
Rust

PUBLISHED BY:
Packt Publishing

PUBLICATION DATE:
February 2020

Start

Scripting languages will provide safety, but not concurrency and speed, while traditional systems programming languages such as C and C++ will definitely give you speed and some concurrency, but forget about safety! If you need safety, concurrency, and speed, then Rust is the only viable option.

In this course, you will learn how Rust guarantees memory and thread safety at compile-time, yet uses zero-cost abstractions without the runtime overhead of a garbage collector. You'll learn how to monitor the flow of data through a pipeline by building your own middleware utility. You'll learn how to utilize I/O to interact with the command line, work with standard library mpsc channels to perform data flows, and create an ergonomic timer for your project. You'll apply key concepts in every section while creating your own middleware tool in Rust along the way.

By the end of this practical course, you will feel comfortable designing safe, consistent, parallel, and high-performance applications in Rust using systems programming.

This course should appeal to intermediate Linux and general Unix programmers, network programmers, and C/C++ programmers interested in learning different approaches to concurrency. Prior knowledge of basic programming concepts is required, and a working knowledge of Rust is assumed.

What You Will Learn

- Explore Rust s package registry for discovering and sharing code
- Use multithreading to unlock the power of multiple cores
- Get to know data-flow rate and speed through a pipeline
- Display time-based statistics using stderr
- Build your own middleware project to control the flow of data between two processes
- Best practices and techniques to set up your project for success
- Test and publish your own project on crates.io

Audience

If you are already a software developer and want to learn systems programming in Rust practically and quickly, this course is for you!

This course will greatly appeal to intermediate Linux and general Unix programmers, network programmers, and C / C++ programmers interested in learning different approaches to concurrency. Prior knowledge of basic programming concepts is required. A working knowledge of Rust is assumed.

About The Author

Nathan Stocks: Nathan Stocks has spent the last 20 years working in software development, mostly in the field of backend infrastructure. He fell in love with Rust in 2016 and began teaching it in 2017. For the past several years, he has focused on systems-level programming. He maintained the AVbin audio library (written in C) from 2011-2013. He has had tons of fun learning, using, and teaching Rust at conferences and online. He also loves Python and PostgreSQL, and still occasionally suffers from nightmares about C and C++. He currently works at GitHub, managing the Git Storage team, and spends his nights in pursuit of someday creating a successful indie game in Rust.

About the Publisher

Packt helps real-world developers put software to work in new ways with over 7,500 practical books and videos covering over 1,000 technologies. With coverage ranging from introductory programming to new and emerging technologies, as well as expert advice from industry-leading figures, Packt has great content for any technical professional looking to stay relevant and keep their skills up-to-date.

More about Packt Publishing

Resources

Supplemental Content