

Read Online Hands On Concurrency With Rust Confidently Build Memory Safe Parallel And Efficient Software In Rust

to a client over a network Parse JSON and binary data using parser combinators such as nom Write an HTTP client that talks to the server using reqwest Modify an existing Rust HTTP server and add SSL to it Master asynchronous programming support in Rust Use external packages in a Rust project Who this book is for This book is for software developers who want to write networking software with Rust. A basic familiarity with networking concepts is assumed. Beginner-level knowledge of Rust will help but is not necessary.

Get familiar with writing programs in the trending new systems programming language that brings together the powerful performance of low-level languages with the advanced features like thread safety in multi-threaded code Key Features Learn the semantics of Rust, which can be significantly different from other programming languages Understand clearly how to work with the Rust compiler which strictly enforces rules that may not be obvious Examples and insights beyond the Rust documentation Book Description Rust is an emerging programming language applicable to areas such as embedded programming, network programming, system programming, and web development. This book will take you from the basics of Rust to a point where your code compiles and does what you intend it to do! This book starts with an introduction to Rust and how to get set for programming, including the rustup and cargo tools for managing a Rust installation and development workflow. Then you'll learn about the fundamentals of structuring a Rust program, such as functions, mutability, data structures, implementing behavior for types, and many more. You will also learn about concepts that Rust handles differently from most other languages. After understanding the Basics of Rust programming, you will learn about the core ideas, such as variable ownership, scope, lifetime, and borrowing. After these key ideas, you will explore making decisions in Rust based on data types by learning about match and if let expressions. After that, you'll work with different data types in Rust, and learn about memory management and smart pointers. What you will learn Install Rust and write your first program with it Understand ownership in Rust Handle different data types Make decisions by pattern matching Use smart pointers Use generic types and type specialization Write code that works with many data types Tap into the standard library Who this book is for This book is for people who are new to Rust, either as their first programming language or coming to it from somewhere else. Familiarity with computer programming in any other language will be helpful in getting the best out of this book.

"This book provides research into parallel & distributed computing, high performance computing, and Grid computing"--Provided by publisher.

Become proficient in designing, developing and deploying effective software systems using the advanced constructs of Rust Key Features Improve your productivity using the latest version of Rust and write simpler and easier code Understand Rust's immutability and ownership principle, expressive type system, safe concurrency Deep dive into the

Read Online Hands On Concurrency With Rust Confidently Build Memory Safe Parallel And Efficient Software In Rust

new domains of Rust like WebAssembly, Networking and Command line tools

Book Description Rust is an empowering language that provides a rare combination of safety, speed, and zero-cost abstractions. Mastering Rust - Second Edition is filled with clear and simple explanations of the language features along with real-world examples, showing you how you can build robust, scalable, and reliable programs. This second edition of the book improves upon the previous one and touches on all aspects that make Rust a great language. We have included the features from latest Rust 2018 edition such as the new module system, the smarter compiler, helpful error messages, and the stable procedural macros. You'll learn how Rust can be used for systems programming, network programming, and even on the web. You'll also learn techniques such as writing memory-safe code, building idiomatic Rust libraries, writing efficient asynchronous networking code, and advanced macros. The book contains a mix of theory and hands-on tasks so you acquire the skills as well as the knowledge, and it also provides exercises to hammer the concepts in. After reading this book, you will be able to implement Rust for your enterprise projects, write better tests and documentation, design for performance, and write idiomatic Rust code.

What you will learn

- Write generic and type-safe code by using Rust's powerful type system
- How memory safety works without garbage collection
- Know the different strategies in error handling and when to use them
- Learn how to use concurrency primitives such as threads and channels
- Use advanced macros to reduce boilerplate code
- Create efficient web applications with the Actix-web framework
- Use Diesel for type-safe database interactions in your web application

Who this book is for The book is aimed at beginner and intermediate programmers who already have familiarity with any imperative language and have only heard of Rust as a new language. If you are a developer who wants to write robust, efficient and maintainable software systems and want to become proficient with Rust, this book is for you. It starts by giving a whirlwind tour of the important concepts of Rust and covers advanced features of the language in subsequent chapters using code examples that readers will find useful to advance their knowledge.

Get to grips with modern software demands by learning the effective uses of Rust's powerful memory safety.

Key Features

- Learn and improve the sequential performance characteristics of your software
- Understand the use of operating system processes in a high-scale concurrent system
- Learn of the various coordination methods available in the Standard library

Book Description Most programming languages can really complicate things, especially with regard to unsafe memory access. The burden on you, the programmer, lies across two domains: understanding the modern machine and your language's pain-points. This book will teach you to how to manage program performance on modern machines and build fast, memory-safe, and concurrent software in Rust. It starts with the fundamentals of Rust and discusses machine architecture concepts. You will be taken through ways to measure and improve the performance of Rust code systematically and how to write collections with confidence. You will learn about the Sync and Send traits applied to

Read Online Hands On Concurrency With Rust Confidently Build Memory Safe Parallel And Efficient Software In Rust

threads, and coordinate thread execution with locks, atomic primitives, data-parallelism, and more. The book will show you how to efficiently embed Rust in C++ code and explore the functionalities of various crates for multithreaded applications. It explores implementations in depth. You will know how a mutex works and build several yourself. You will master radically different approaches that exist in the ecosystem for structuring and managing high-scale systems. By the end of the book, you will feel comfortable with designing safe, consistent, parallel, and high-performance applications in Rust. What you will learn Probe your programs for performance and accuracy issues Create your own threading and multi-processing environment in Rust Use coarse locks from Rust's Standard library Solve common synchronization problems or avoid synchronization using atomic programming Build lock-free/wait-free structures in Rust and understand their implementations in the crates ecosystem Leverage Rust's memory model and type system to build safety properties into your parallel programs Understand the new features of the Rust programming language to ease the writing of parallel programs Who this book is for This book is aimed at software engineers with a basic understanding of Rust who want to exploit the parallel and concurrent nature of modern computing environments, safely.

Explore the support Rust offers for creating functional applications in Rust. Learn about various design patterns, implementing concurrency, metaprogramming, and so on in the process Key Features Learn generics, organization, and design patterns in functional programming Modularize your applications and make them highly reusable and testable using functional design patterns Get familiar with complex concepts such as metaprogramming, concurrency, and immutability Book Description Functional programming allows developers to divide programs into smaller, reusable components that ease the creation, testing, and maintenance of software as a whole. Combined with the power of Rust, you can develop robust and scalable applications that fulfill modern day software requirements. This book will help you discover all the Rust features that can be used to build software in a functional way. We begin with a brief comparison of the functional and object-oriented approach to different problems and patterns. We then quickly look at the patterns of control flow, data the abstractions of these unique to functional programming. The next part covers how to create functional apps in Rust; mutability and ownership, which are exclusive to Rust, are also discussed. Pure functions are examined next and you'll master closures, their various types, and currying. We also look at implementing concurrency through functional design principles and metaprogramming using macros. Finally, we look at best practices for debugging and optimization. By the end of the book, you will be familiar with the functional approach of programming and will be able to use these techniques on a daily basis. What you will learn How Rust supports the use of basic functional programming principles Use functional programming to handle concurrency with elegance Read and interpret complex type signatures for types and functions Implement powerful abstractions using meta programming in Rust Create quality

Read Online Hands On Concurrency With Rust Confidently Build Memory Safe Parallel And Efficient Software In Rust

intensive and low-level systems applications. This book will give you a head start to solve systems programming and application tasks with Rust. We start off with an argumentation of Rust's unique place in today's landscape of programming languages. You'll install Rust and learn how to work with its package manager Cargo. The various concepts are introduced step by step: variables, types, functions, and control structures to lay the groundwork. Then we explore more structured data such as strings, arrays, and enums, and you'll see how pattern matching works. Throughout all this, we stress the unique ways of reasoning that the Rust compiler uses to produce safe code. Next we look at Rust's specific way of error handling, and the overall importance of traits in Rust code. The pillar of memory safety is treated in depth as we explore the various pointer kinds. Next, you'll see how macros can simplify code generation, and how to compose bigger projects with modules and crates. Finally, you'll discover how we can write safe concurrent code in Rust and interface with C programs, get a view of the Rust ecosystem, and explore the use of the standard library. Style and approach The book takes a pragmatic approach, showing various methods to solve systems programming tasks with Rust and develop resource intensive and low-level systems applications.

This practical book introduces systems programmers to Rust, the new and cutting-edge language that's still in the experimental/lab stage. You'll learn how Rust offers the rare and valuable combination of statically verified memory safety and low-level control—imagine C++, but without dangling pointers, null pointer dereferences, leaks, or buffer overruns. Author Jim Blandy—the maintainer of GNU Emacs and GNU Guile—demonstrates how Rust has the potential to be the first usable programming language that brings the benefits of an expressive modern type system to systems programming. Rust's rules for borrowing, mutability, ownership, and moves versus copies will be unfamiliar to most systems programmers, but they're key to Rust's unique advantages. This book presents Rust's rules clearly and economically; elaborates on their consequences; and shows you how to express the programs you want to write in terms that Rust can prove are free of a broad class of common errors.

Many of the normal concerns faced by application developers are amplified by the challenges of web-scale concurrency, real-time performance expectations, multi-core support, and efficiently consuming services without constantly managing I/O blocks. Although it's possible to solve most of these issues with existing languages and frameworks, Go is designed to handle them right out of the box, making for a more natural and productive coding experience. Developed at Google for its own internal use, Go now powers dozens of nimble startups, along with name brands like Canonical, Heroku, SoundCloud, and Mozilla, who rely on highly performant services for their infrastructure. Go in Action introduces the unique features and concepts of the Go language, guiding readers from inquisitive developers to Go gurus. It provides hands-on experience with writing real-world applications including web sites and network servers, as well as techniques to manipulate and convert data at incredibly high speeds. It also goes in-depth with the language and explains the tricks and secrets that the Go masters are using to make their applications perform. For example, it looks at Go's powerful reflection libraries and uses real-world examples of integration with C code. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications.

This two-volume set LNCS 12269 and LNCS 12270 constitutes the refereed proceedings of the 16th International Conference on Parallel Problem Solving from Nature, PPSN 2020, held in Leiden, The Netherlands, in September 2020. The 99 revised full papers were carefully reviewed and selected from 268 submissions. The topics cover classical subjects such as automated algorithm selection and configuration; Bayesian- and surrogate-assisted optimization; benchmarking and performance measures; combinatorial optimization; connection between nature-inspired optimization and artificial intelligence; genetic and evolutionary algorithms; genetic programming; landscape analysis;

Read Online Hands On Concurrency With Rust Confidently Build Memory Safe Parallel And Efficient Software In Rust

multiobjective optimization; real-world applications; reinforcement learning; and theoretical aspects of nature-inspired optimization.

An insightful guide to learning the Go programming language About This Book Get insightful coverage of Go programming syntax, constructs, and idioms to help you understand Go code Get a full explanation of all the known GoF design patterns in Go, including comprehensive theory and examples Learn to apply the nuances of the Go language, and get to know the open source community that surrounds it to implement a wide range of start-up quality projects Who This Book Is For Beginners to Go who are comfortable in other OOP languages like Java, C#, or Python will find this course interesting and beneficial. What You Will Learn Install and configure the Go development environment to quickly get started with your first program Use the basic elements of the language including source code structure, variables, constants, and control flow primitives Get to know all the basic syntax and tools you need to start coding in Go Create unique instances that cannot be duplicated within a program Build quirky and fun projects from scratch while exploring patterns, practices, and techniques, as well as a range of different technologies Create websites and data services capable of massive scaling using Go's net/http package, Explore RESTful patterns as well as low-latency WebSocket APIs Interact with a variety of remote web services to consume capabilities, ranging from authentication and authorization to a fully functioning thesaurus In Detail The Go programming language has firmly established itself as a favorite for building complex and scalable system applications. Go offers a direct and practical approach to programming that lets programmers write correct and predictable code using concurrency idioms and a full-featured standard library. This practical guide is full of real-world examples to help you get started with Go in no time at all. You'll start by understanding the fundamentals of Go, then get a detailed description of the Go data types, program structures, and Maps. After that, you'll learn how to use Go concurrency idioms to avoid pitfalls and create programs that are exact in expected behavior. Next, you will get familiar with the tools and libraries that are available in Go to write and exercise tests, benchmarking, and code coverage. After that, you will be able to utilize some of the most important features of GO such as Network Programming and OS integration to build efficient applications. Then you'll start applying your skills to build some amazing projects in Go. You will learn to develop high-quality command-line tools that utilize the powerful shell capabilities and perform well using Go's built-in concurrency mechanisms. Scale, performance, and high availability lie at the heart of our projects, and the lessons learned throughout the sections will arm you with everything you need to build world-class solutions. You will get a feel for app deployment using Docker and Google App Engine. Each project could form the basis of a start-up, which means they are directly applicable to modern software markets. With these skills in hand, you will be able to conquer all your fears of application development and go on to build large, robust and succinct apps in Go. This Learning Path combines some of the best that Packt has to offer in one complete, curated package. It includes content from the following Packt products: Learning Go Programming Go Design Patterns Go Programming Blueprints, Second Edition Style and approach Full of real-world, practical examples, this course teaches you the widely used design patterns and best practices in Go in a step-by-step manner. It also provides fun projects that involve building applications from scratch.

Information Technology and Product Development: A Research Agenda presents important new research from varied disciplines aimed at developing new theoretical concepts and insights on the application of IT in product and service innovation. Drawing on the work of researchers in such varied management areas as information services, technology management, marketing, operations, business strategy and organizational behavior, the book redefines the role of IT in product and service development and the organizational and management issues underlying the successful deployment of IT in innovation contexts, and provides a

Read Online Hands On Concurrency With Rust Confidently Build Memory Safe Parallel And Efficient Software In Rust

foundation for future research on the diverse types of IT applications in product development and their potential impact on both product and service innovation. Reflecting two critical shifts in the service sector – the increased complexity and convergence in products and services, along with the rise of the Internet and rapid digitization of products and services – the book is organized into three sections. Section 1 presents four chapters that focus on the traditional areas of project and process management; Section 2 presents four chapters focusing on the emerging areas of collaborative innovation and knowledge co-creation; and Section 3 presents one chapter that draws it all together and identifies some of the important themes and issues for future research. This important new work has much to offer academic researchers in management in its in-depth theoretical analysis of the wide range of organizational and management issues associated with the application of IT in product and service development. It will also appeal to researchers and thought-leaders in consulting organizations whose primary area of interest is product development or IT applications.

Discover the powerful, hidden features of Rust you need to build robust, concurrent, and fast applications About This Book Learn how concurrency works in Rust and why it is safe Get to know the different philosophies of error handling and how to use them wisely After reading this book, you will be able to migrate your legacy C or C++ application to a Rust environment Who This Book Is For The target audience would be readers having knowledge of other programming languages and are able to work fluently in the operating system of their choice, be it Linux, OS X or Windows. Since Rust is a rather new language, they are interested in programming beyond simply using it for work. The book focuses on intermediate and advanced features of Rust. What You Will Learn Implement unit testing patterns with the standard Rust tools Get to know the different philosophies of error handling and how to use them wisely Appreciate Rust's ability to solve memory allocation problems safely without garbage collection Get to know how concurrency works in Rust and use concurrency primitives such as threads and message passing Use syntax extensions and write your own Create a Web application with Rocket Use Diesel to build safe database abstractions In Detail If concurrent programs are giving you sleepless nights, Rust is your go-to language. Being one of the first ever comprehensive books on Rust, it is filled with real-world examples and explanations, showing you how you can build scalable and reliable programs for your organization. We'll teach you intermediate to advanced level concepts that make Rust a great language. Improving performance, using generics, building macros, and working with threads are just some of the topics we'll cover. We'll talk about the official toolsets and ways to discover more. The book contains a mix of theory interspersed with hands-on tasks, so you acquire the skills as well as the knowledge. Since programming cannot be learned by just reading, we provide exercises (and solutions) to hammer the concepts in. After reading this book, you will be able to implement Rust for your enterprise project, deploy the software, and will know the best practices of coding in Rust. Style and approach This book is your one stop guide to the Rust programming language and covers advanced-level concepts in a detailed manner using real-world examples.

The official book on the Rust programming language, written by the Rust development team at the Mozilla Foundation, fully updated for Rust 2018. The Rust Programming Language is the official book on Rust: an open source systems programming

Read Online Hands On Concurrency With Rust Confidently Build Memory Safe Parallel And Efficient Software In Rust

language that helps you write faster, more reliable software. Rust offers control over low-level details (such as memory usage) in combination with high-level ergonomics, eliminating the hassle traditionally associated with low-level languages. The authors of The Rust Programming Language, members of the Rust Core Team, share their knowledge and experience to show you how to take full advantage of Rust's features--from installation to creating robust and scalable programs. You'll begin with basics like creating functions, choosing data types, and binding variables and then move on to more advanced concepts, such as:

- Ownership and borrowing, lifetimes, and traits
- Using Rust's memory safety guarantees to build fast, safe programs
- Testing, error handling, and effective refactoring
- Generics, smart pointers, multithreading, trait objects, and advanced pattern matching
- Using Cargo, Rust's built-in package manager, to build, test, and document your code and manage dependencies
- How best to use Rust's advanced compiler with compiler-led programming techniques

You'll find plenty of code examples throughout the book, as well as three chapters dedicated to building complete projects to test your learning: a number guessing game, a Rust implementation of a command line tool, and a multithreaded server. New to this edition: An extended section on Rust macros, an expanded chapter on modules, and appendixes on Rust development tools and editions.

Identifies Recent Technological Developments Worldwide The field of grid computing has made rapid progress in the past few years, evolving and developing in almost all areas, including concepts, philosophy, methodology, and usages. **Grid Computing: Infrastructure, Service, and Applications** reflects the recent advances in this field, covering the research aspects that involve infrastructure, middleware, architecture, services, and applications. **Grid Systems Across the Globe** The first section of the book focuses on infrastructure and middleware and presents several national and international grid systems. The text highlights China Research and Development environment Over Wide-area Network (CROWN), several ongoing cyberinfrastructure efforts in New York State, and Enabling Grids for E-science (EGEE), which is co-funded by the European Commission and the world's largest multidisciplinary grid infrastructure today. The second part of the book discusses recent grid service advances. The authors examine the UK National Grid Service (NGS), the concept of resource allocation in a grid environment, OMIIBPEL, and the possibility of treating scientific workflow issues using techniques from the data stream community. The book describes an SLA model, reviews portal and workflow technologies, presents an overview of PKIs and their limitations, and introduces PIndex, a peer-to-peer model for grid information services. **New Projects and Initiatives** The third section includes an analysis of innovative grid applications. Topics covered include the WISDOM initiative, incorporating flow-level networking models into grid simulators, system-level virtualization, grid usage in the high-energy physics environment in the LHC project, and the Service Oriented HLA RTI (SOHR) framework. With a comprehensive summary of past advances, this text is a window into the future of this nascent technology, forging a path for the next generation of cyberinfrastructure developers.

This three-volume book highlights significant advances in the development of new information systems technologies and architectures. Further, it helps readers solve specific research and analytical problems and glean useful knowledge and business value from data. Each chapter provides an analysis of a specific technical problem, followed by a numerical analysis, simulation,

and implementation of the solution to the real-world problem. Managing an organization, especially in today's rapidly changing environment, is a highly complex process. Increased competition in the marketplace, especially as a result of the massive and successful entry of foreign businesses into domestic markets, changes in consumer behaviour, and broader access to new technologies and information, calls for organisational restructuring and the introduction and modification of management methods using the latest scientific advances. This situation has prompted various decision-making bodies to introduce computer modelling of organization management systems. This book presents the peer-reviewed proceedings of the 40th Anniversary International Conference "Information Systems Architecture and Technology" (ISAT), held on September 15–17, 2019, in Wrocław, Poland. The conference was organised by the Computer Science Department, Faculty of Computer Science and Management, Wrocław University of Sciences and Technology, and University of Applied Sciences in Nysa, Poland. The papers have been grouped into three major sections: Part I—discusses topics including, but not limited to, artificial intelligence methods, knowledge discovery and data mining, big data, knowledge-based management, Internet of Things, cloud computing and high-performance computing, distributed computer systems, content delivery networks, and service-oriented computing. Part II—addresses various topics, such as system modelling for control, recognition and decision support, mathematical modelling in computer system design, service-oriented systems, and cloud computing, and complex process modelling. Part III—focuses on a number of themes, like knowledge-based management, modelling of financial and investment decisions, modelling of managerial decisions, production systems management, and maintenance, risk management, small business management, and theories and models of innovation.

Fourth Edition (Traditional Chinese Translation) Sheds New Light on Open Source Intelligence Collection and Analysis. Author Michael Bazzell has been well known and respected in government circles for his ability to locate personal information about any target through Open Source Intelligence (OSINT). In this book, he shares his methods in great detail. Each step of his process is explained throughout sixteen chapters of specialized websites, application programming interfaces, and software solutions. Based on his live and online video training at IntelTechniques.com, over 250 resources are identified with narrative tutorials and screen captures. This book will serve as a reference guide for anyone that is responsible for the collection of online content. It is written in a hands-on style that encourages the reader to execute the tutorials as they go. The search techniques offered will inspire analysts to "think outside the box" when scouring the internet for personal information. Much of the content of this book has never been discussed in any publication. Always thinking like a hacker, the author has identified new ways to use various technologies for an unintended purpose. This book will improve anyone's online investigative skills. Among other techniques, you will learn how to locate: Hidden Social Network Content Cell Phone Owner Information Twitter GPS & Account Data Hidden Photo GPS & Metadata Deleted Websites & Posts Website Owner Information Alias Social Network Profiles Additional User Accounts Sensitive Documents & Photos Live Streaming Social Content IP Addresses of Users Newspaper Archives & Scans Social Content by Location Private Email Addresses Hidden Personal Videos Duplicate Copies of Photos Personal Radio Communications Compromised Email Information Wireless Routers by Location Hidden Mapping Applications Complete Facebook Data Free

Read Online Hands On Concurrency With Rust Confidently Build Memory Safe Parallel And Efficient Software In Rust

Investigative Software Alternative Search Engines Mobile App Network Data Unlisted Addresses Unlisted Phone Numbers Useful Browser Extensions Public Government Records Document Metadata Rental Vehicle Contracts Online Criminal Activity Hands-On Data Structures and Algorithms with Rust will help you in upgrading your earlier knowledge of Rust so that you shift to a confident developer by implementing the algorithms in a practical environment. This would be an essential reference guide for end-user/reader to understand the fundamental techniques of Rust. This guide will cover ...

A comprehensive guide in developing and deploying high performance microservices with Rust Key Features Start your microservices journey and get a broader perspective on microservices development using RUST 2018, Build, deploy, and test microservices using AWS Explore advanced techniques for developing microservices such as actor model, Requests Routing, and threads Book Description Microservice architecture is sweeping the world as the de facto pattern for building web-based applications. Rust is a language particularly well-suited for building microservices. It is a new system programming language that offers a practical and safe alternative to C. This book describes web development using the Rust programming language and will get you up and running with modern web frameworks and crates with examples of RESTful microservices creation. You will deep dive into Reactive programming, and asynchronous programming, and split your web application into a set of concurrent actors. The book provides several HTTP-handling examples with manageable memory allocations. You will walk through stateless high-performance microservices, which are ideally suitable for computation or caching tasks, and look at stateful microservices, which are filled with persistent data and database interactions. As we move along, you will learn how to use Rust macros to describe business or protocol entities of our application and compile them into native structs, which will be performed at full speed with the help of the server's CPU. Finally, you will be taken through examples of how to test and debug microservices and pack them into a tiny monolithic binary or put them into a container and deploy them to modern cloud platforms such as AWS. What you will learn Get acquainted with leveraging Rust web programming Get to grips with various Rust crates, such as hyper, Tokio, and Actix Explore RESTful microservices with Rust Understand how to pack Rust code to a container using Docker Familiarize yourself with Reactive microservices Deploy your microservices to modern cloud platforms such as AWS Who this book is for This book is for developers who have basic knowledge of RUST, and want to learn how to build, test, scale, and manage RUST microservices. No prior experience of writing microservices in RUST is assumed.

This volume constitutes the proceedings of the 19th Asia Simulation Conference, AsiaSim 2019, held in Singapore, Singapore, in October 2019. The 19 revised full papers and 5 short papers presented in this volume were carefully reviewed and selected from 36 submissions. The papers are organized in topical sections on simulation and modeling methodology; numerical and Monte Carlo simulation; simulation applications: blockchain, deep learning and cloud; simulation and visualization; simulation applications; short papers.

Rust in Action introduces the Rust programming language by exploring numerous systems programming concepts and techniques. You'll be learning Rust by delving into how computers work under the hood. You'll find yourself playing with persistent storage, memory, networking and even tinkering with CPU instructions. The book takes you through using Rust to extend other applications and teaches you tricks to write blindingly fast code. You'll also discover parallel and concurrent programming. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications.

The Rust Programming Language is the official book on Rust, an open-source, community-developed systems programming language that

Read Online Hands On Concurrency With Rust Confidently Build Memory Safe Parallel And Efficient Software In Rust

runs blazingly fast, prevents segfaults, and guarantees thread safety. This is the undisputed go-to guide to Rust, written by two members of the Rust core team, with feedback and contributions from 42 members of the community. The book assumes that you've written code in another programming language but makes no assumptions about which one, meaning the material is accessible and useful to developers from a wide variety of programming backgrounds. Known by the Rust community as "The Book," The Rust Programming Language includes concept chapters, where you'll learn about a particular aspect of Rust, and project chapters, where you'll apply what you've learned so far to build small programs. The Book opens with a quick hands-on project to introduce the basics then explores key concepts in depth, such as ownership, the type system, error handling, and fearless concurrency. Next come detailed explanations of Rust-oriented perspectives on topics like pattern matching, iterators, and smart pointers, with concrete examples and exercises--taking you from theory to practice. The Rust Programming Language will also show you how to:

- Grasp important concepts unique to Rust, like ownership, borrowing, and lifetimes
- Use Cargo, Rust's built-in package manager, to build and maintain your code, including downloading and building dependencies
- Effectively use Rust's zero-cost abstractions and employ your own

You'll learn to develop reliable code that's speed and memory efficient, while avoiding the infamous and arcane programming pitfalls common at the systems level. When you need to dive down into lower-level control, this guide will show you how without taking on the customary risk of crashes or security holes and without requiring you to learn the fine points of a fickle toolchain. You'll also learn how to create command line programs, build single- and multithreaded web servers, and much more. The Rust Programming Language fully embraces Rust's potential to empower its users. This friendly and approachable guide will help you build not only your knowledge of Rust but also your ability to program with confidence in a wider variety of domains.

????????????,???UNIX???C????????????,????????????????IT????????????UNIX????????????

Take advantage of Rust to ensure safety and concurrency About This Video Quickly learn the essentials of systems programming in Rust and learn to monitor data flows through a pipeline Dive into a fast-paced and informative course which includes hands-on implementations of important concepts in every section Work with the latest version of Rust and develop a utility that's compatible with Linux, MacOS, and Windows In Detail 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 mpvc 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.

Go represents an attempt to improve on some weaknesses of traditional compiled programming languages. It de-emphasizes or removes error-prone language structures like class inheritance and exception handling. It features great concurrency support and automatic memory management (garbage collection). Existing Go books tend to be highly technical in nature, teaching all aspects of the language regardless of their relevance to beginners. This book, rather than talking about the features of Go in abstract terms, features simple, clear examples that

Read Online Hands On Concurrency With Rust Confidently Build Memory Safe Parallel And Efficient Software In Rust

demonstrate Go in action, and diagrams to explain difficult concepts. This book will not only teach developers basic language features, it will get them comfortable consulting error output, documentation, and search engines to find solutions to problems. It will teach all the conventions and techniques that employers expect an entry-level Go developer to know.

[Copyright: edee7245da93296b6eaf4c46ff3f1fc1](#)