1. Record Nr. UNINA9911009184803321

Autore	Nellaiyapen Sendil Kumar
Titolo	PRACTICAL WEBASSEMBLY : explore the fundamentals of webassembly programming using rust
Pubbl/distr/stampa	[Place of publication not identified] : , : PACKT PUBLISHING LIMITED, , 2022
ISBN	9781838828004 1838828001
Descrizione fisica	1 online resource (232 p.)
Disciplina	006.7/6
Soggetti	Web site development - Computer programs Application software - Development Internet programming Assembly languages (Electronic computers) Rust (Computer program language)
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Intro -- Title page -- Copyright and Credits -- Dedication -- Contributors -- Table of Contents -- Preface -- Section 1: Introduction to WebAssembly -- Chapter 1: Understanding LLVM -- Technical requirements -- Understanding compilers -- Compiled languages -- Compiler efficiency -- Exploring LLVM -- LLVM in action -- Summary -- Chapter 2: Understanding Emscripten -- Technical requirements -- Installing Emscripten using emsdk -- Generating asm.js using Emscripten -- Running Hello World with Emscripten in Node.js -- Running Hello World with Emscripten in the browser Exploring other options in emsdk -- Listing the tools and SDK -- Managing the tools and SDK -- Understanding various levels of optimizations -- Optimizations -- Closure Compiler -- Summary -- Chapter 3: Exploring WebAssembly Modules -- Technical requirements -- Understanding how WebAssembly works -- Understanding JavaScript execution inside the JavaScript engine -- Understanding WebAssembly execution inside the JavaScript engine -- Exploring the WebAssembly text format -- Building a function in WebAssembly text

format -- Summary -- Section 2: WebAssembly Tools
Chapter 4: Understanding WebAssembly Binary Toolkit -- Technical requirements -- Getting started with WABT -- Installing WABT -- Converting WAST into WASM -- Converting WASM into WAST -- f or -- fold-exprs -- Converting WASM into C -- simple.h -- simple.c -- Converting WAST into JSON -- Understanding a few other tools provided by WABT -- wasm-objdump -- wasm-strip -- wasm-validate -- wasm-interp -- Summary -- Chapter 5: Understanding Sections in WebAssembly Modules -- Technical requirements -- Exports and imports -- Exports -- Imports -- Globals -- Start -- Memory -- Summary
Chapter 6: Installing and Using Binaryen -- Technical requirements -- Installing and using Binaryen -- Linux/macOS -- Windows -- wasm-as -- wasm-dis -- wasm-opt -- wasm2js -- Summary -- Section 3: Rust and WebAssembly -- Chapter 7: Integrating Rust with WebAssembly -- Technical requirements -- Installing Rust -- Converting Rust into WebAssembly via rustc -- Converting Rust into WebAssembly via Cargo -- Installing wasm-bindgen -- Converting Rust into WebAssembly via wasm-bindgen -- Summary -- Chapter 8: Bundling WebAssembly Using wasm-pack -- Technical requirements
Bundling WebAssembly modules with webpack -- Bundling WebAssembly modules with Parcel -- Introducing wasm-pack -- Why do you need wasm-pack? -- How to use wasm-pack -- Packing and publishing using wasm-pack -- Summary -- Chapter 9: Crossing the Boundary between Rust and WebAssembly -- Technical requirements -- Sharing classes from Rust with JavaScript -- Sharing classes from JavaScript with Rust -- Calling the JavaScript API via WebAssembly -- Calling closures via WebAssembly -- Importing the JavaScript function into Rust -- Calling a web API via WebAssembly -- Summary

Sommario/riassunto

Understand the basic building blocks of WebAssembly and learn, install, and use various tools from the Rust and WebAssembly ecosystemKey FeaturesUnderstand the Rust programming language and WebAssembly concepts for web developmentBuild web, mobile, and embedded apps using WebAssemblyEnhance the scalability and resilience of your web appsBook DescriptionRust is an open source language tuned toward safety, concurrency, and performance. WebAssembly brings all the capabilities of the native world into the JavaScript world. Together, Rust and WebAssembly provide a way to create robust and performant web applications. They help make your web applications blazingly fast and have small binaries. Developers working with JavaScript will be able to put their knowledge to work with this practical guide to developing faster and maintainable code. Complete with step-by-step explanations of essential concepts, examples, and self-assessment questions, you'll begin by exploring WebAssembly, using the various tools provided by the ecosystem, and understanding how to use WebAssembly and JavaScript together to build a high-performing application. You'll then learn binary code to work with a variety of tools that help you to convert native code into WebAssembly. The book will introduce you to the world of Rust and the ecosystem that makes it easy to build/ship WebAssembly-based applications. By the end of this WebAssembly Rust book, you'll be able to create and ship your own WebAssembly applications using Rust and JavaScript, understand how to debug, and use the right tools to optimize and deliver high-performing applications.What you will learnExplore WebAssembly and the different tools available in the WebAssembly ecosystemUnderstand the raw WebAssembly binary and the WebAssembly text formatUse the Web and JavaScript API with wasm-bindgenOptimize Rust and WebAssembly for high

performanceRun and debug WebAssembly and Rust codeExplore various tools available in the RustWASM ecosystemWho this book is forThis book is for JavaScript developers who want to deliver better performance and ship type-safe code. Rust developers or backend engineers looking to build full-stack applications without worrying too much about JavaScript programming will also find the book useful.