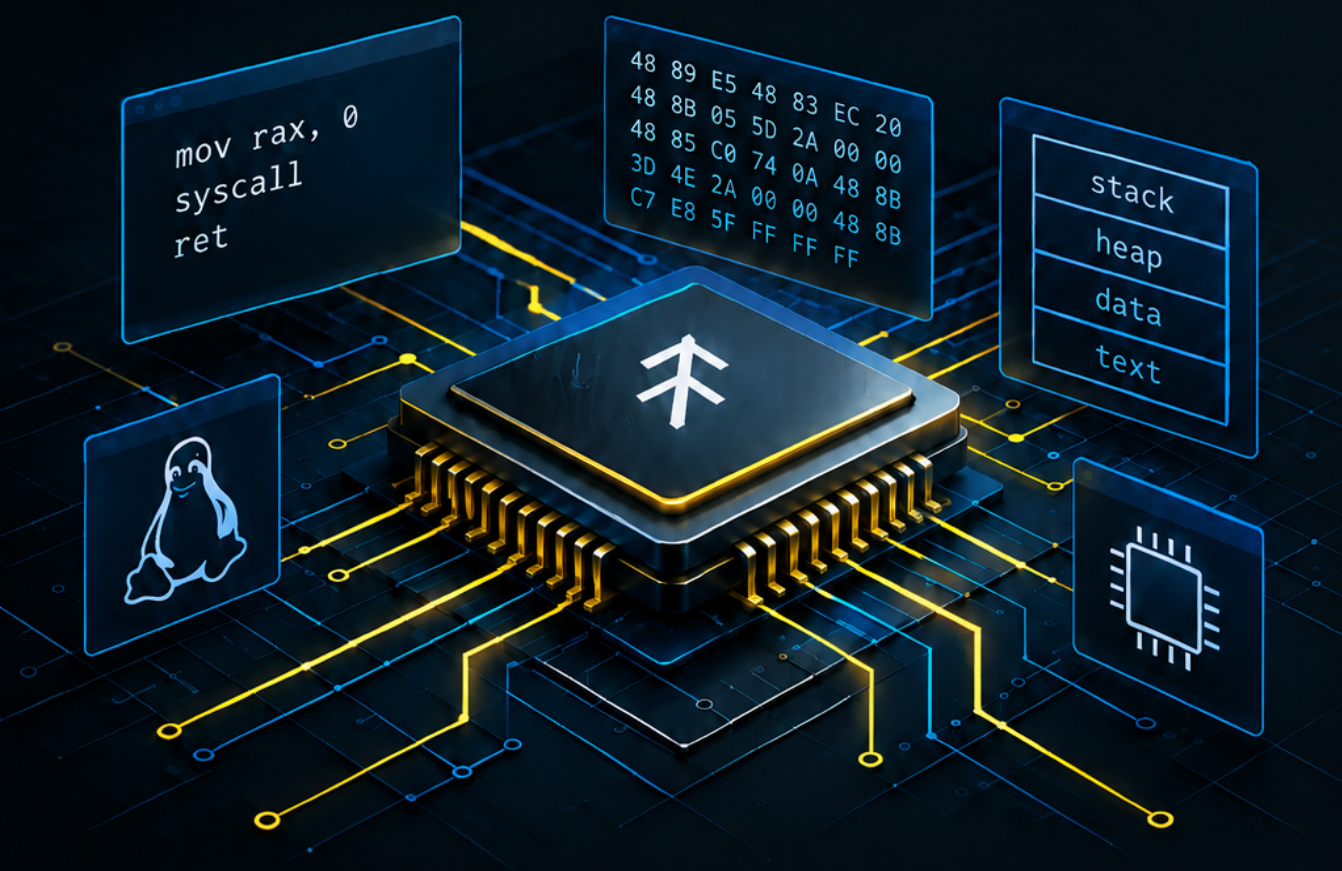


ODIN

A PRACTICAL GUIDE TO LOW-LEVEL PROGRAMMING



FINN CALLAHAN & DAVE MIGUEL

Odin: A Practical Guide to Low-Level Programming

Design, Build, and Optimize Systems Software in Odin

Steve Publications

This book is available at

<https://leanpub.com/odinapracticalguidetolow-levelprogramming>

This version was published on 2026-10-01



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2026 Steve Publications

Contents

Design, Build and Optimize Systems Software in Odin	1
Introduction	2
What This Book Is	2
What This Book Is Not	3
Who This Book Is For	3
Prerequisites	4
How to Read This Book	5
Technical Conventions	5
Acknowledging the Ecosystem	6
What You Will Know After This Book	6
Chapter 1: Why Odin? The Design Philosophy	8
The State of Systems Programming in 2026	8
What Odin Is and Is Not	9
Data-Oriented Design as First-Class Concern	10
The Pragmatist's Manifesto: Simplicity Over Power	12
Comparison Landscape: C, C++, Rust, Go, Zig	13
The Odin Promise	16
Chapter 2: Setting Up the Development Environment	18
Installing the Odin Compiler	18
Building from Source	20
Your First Program: Hellope World	21
The odin Command: Build, Run, Test, Check	23
Editor and Language Server Setup with OLS	25
Project Structure and Collections	28
Verifying Your Setup	30
Chapter 3: Language Fundamentals: Declarations, Types and Values . .	32
Statements, Expressions and Automatic Semicolon Insertion	32
Variables: Declaration, Initialization and Zeroing	32

CONTENTS

Constants and Compile-Time Expressions	33
Primitive Types: Integers, Floats, Booleans, Runes	33
Distinct Types vs Aliases: Type Identity Matters	35
Literals: Numbers, Strings and the Underscore Convention	35
Summary	36
Chapter 4: Procedures: Functions with Intent	37
Declaring and Calling Procedures	37
Parameters, Named Results and Multiple Return Values	37
Calling Conventions: <code>odin</code> , <code>c</code> , <code>contextless</code> and <code>Beyond</code>	38
Variadic Procedures and Type Safety	39
Defer and Scope-Guaranteed Cleanup	40
Tail Calls and Recursion	41
Summary	42
Chapter 5: Control Flow Without Hidden Complexity	43
Range-Based For Loops	43
If Statements and Ternary Expressions	44
Switch: Exhaustive, No Fallthrough, Pattern-Match-Free	45
The <code>or</code> Family: <code>or_return</code> , <code>or_else</code> , <code>or_break</code> , <code>or_continue</code>	46
The <code>When</code> Statement: Compile-Time Branching	47
Why No <code>Goto</code> and No Exceptions	48
Summary	48
Chapter 6: Composite Types: Structs, Unions, Enums and Bit Sets	50
Structs: Layout, Alignment, Padding and <code>#packed</code>	50
Unions and <code>#raw_union</code> : Tagged vs Untagged	51
Enums and Enumerated Arrays	52
Bit Sets: Type-Safe Flag Management	53
Fields, Tags and the <code>Using</code> Statement	53
Aggregate Initialization and Partial Initialization	54
Summary	55
Chapter 7: Memory Layout: From Bytes to Caches	56
How the Compiler Lays Out Types	56
Alignment, Packing and ABI Requirements	57
Measuring Size and Offset with <code>sizeof</code> and <code>alignof</code>	57
Cache Lines, Prefetching and Data Locality	58
Struct Layout Optimization Techniques	59
Matching C Layouts: Foreign Interop and Hardware Formats	60

Summary	61
Chapter 8: Arrays, Slices and Dynamic Arrays	62
Fixed Arrays: Stack-Bound and Sized	62
Slices: Pointers with Length	63
Dynamic Arrays: Growth, Allocation and Ownership	64
Slice Operations: Append, Delete and Copy	65
Array Arithmetic and Component-Wise Operations	66
When to Use Each Array Type	67
Summary	68
Chapter 9: Maps and String Handling	69
The Map Type: Hash Table Semantics	69
Map Keys: What Is Allowed	70
Strings as Pointer Plus Length	70
UTF-8, Runes and Character Encoding	71
C Strings and Interop with External APIs	72
String Interning and Pooling	73
Summary	73
Chapter 10: Pointers, Memory Addresses and Ownership	74
Single Pointers: $\wedge$ T Semantics	74
Multi-Pointers: $[\wedge]$ T for Typed Untyped-Length Access	75
Rawptr and Type Erasure	76
Pointer Arithmetic and Unsafe Access	76
Memory Ownership: The Programmer's Responsibility	77
Common Pointer Mistakes and Corruption Patterns	78
Summary	79
Chapter 11: The Context System and Memory Allocators	80
What the Context Is and How It Flows	80
Context Fields: Allocators, Logger, RNG and More	80
new, make, free and delete	81
The Temporary Allocator: Arena-Style Allocation	82
Tracking Allocator: Debugging Memory Issues	83
Custom Allocators and Context Overrides	84
Summary	84
Chapter 12: Parametric Polymorphism: Generic Procedures and Types	86
Explicit vs Implicit Parametric Polymorphism	86

CONTENTS

Where Clauses and Type Constraints	86
Parametric Structs: Generic Data Structures	87
Type Inference and Its Limits	88
Type-Specific Code Paths (No Overloading)	89
Comparison with Rust Traits and C++ Templates	89
Summary	90
Chapter 13: Data-Oriented Design with Odin	91
What Data-Oriented Design Means	91
AoS vs SoA vs AoSoA: Memory Layouts for Performance	91
The #soa Modifier: Odin's Struct of Arrays	92
SIMD Vectors: Native Support and Swizzling	92
Matrix and Quaternion Types	93
Designing Cache-Friendly Data Structures	94
Summary	95
Chapter 14: Error Handling: Explicit by Design	96
Why No Exceptions: Design Rationale	96
Multiple Return Values for Success and Error	97
The or_return Pattern	97
Custom Error Types and Discriminated Unions	98
Error Propagation Strategies	99
Comparison with Go and Rust Approaches	99
Summary	100
Chapter 15: Packages and the Module System	101
Packages: Directories as Units of Compilation	101
Import Names and Aliases	101
Visibility: @(private), #+private and Export Control	102
Core, Vendor and User Collections	103
Foreign Import and Foreign Blocks	104
Structuring Large Projects	105
Summary	106
Chapter 16: Foreign Function Interface: Calling C and the OS	107
Foreign Import: Linking to Libraries	107
Foreign Blocks: Declaring External Symbols	107
Type Mapping: Odin Types to C Types	108
Calling Conventions and ABI Details	110
String Interop: string vs cstring	110

CONTENTS

Working with System Libraries: Windows, POSIX, macOS	111
Summary	112
Chapter 17: File I/O, Networking and the Operating System	113
The New core:os v2: Design and API	113
File Operations: Open, Read, Write, Seek	113
Paths, Directories and Environment	115
Networking with core:net: Sockets and DNS	116
Non-Blocking I/O with core:nbio	117
Resource Lifetimes and Cleanup	117
Summary	118
Chapter 18: Concurrency and Threading	119
Threads and the Operating System	119
Creating and Managing Threads	119
Thread Pools and Task Queues	120
Mutexes, Read-Write Locks and Semaphores	121
Atomic Operations and Lock-Free Programming	122
Context in a Multi-Threaded World	123
Summary	123
Chapter 19: Runtime Type Information and Reflection	124
TypeId: Runtime Type Identifiers	124
The Reflect Package: Inspecting Types	124
Field Tags and Metadata	125
Any Type and Type Erasure	126
Use Cases: Serialization, Debugging and Tooling	127
The Cost of RTTI	128
Summary	128
Chapter 20: Inline Assembly and Intrinsics	130
Intrinsics: Compiler-Hinted Operations	130
SIMD Operations in core:simd	130
The New Inline Asm Template System	131
Writing Portable Assembly	132
When to Use Assembly vs High-Level Code	133
Profiling Assembly Output	134
Summary	135
Chapter 21: Project 1 - A Command-Line Binary File Parser	136

CONTENTS

Requirements and Design Decisions	136
Defining the File Format with Odin Types	136
Parsing Logic and Error Handling	137
Command-Line Argument Processing with core:flags	138
Testing with Synthetic Input Files	138
Packaging and Distribution	139
Summary	140
Chapter 22: Project 2 - A Custom Memory Allocator	141
Allocator Requirements and Interface	141
Implementing an Arena Allocator	141
Implementing a Pool Allocator for Fixed-Size Objects	142
Integration with the Context System	143
Performance Measurement and Comparison	143
Debugging and Validation	144
Summary	144
Chapter 23: Project 3 - A Data-Oriented Physics Simulation	146
Simulation Requirements and Data Model	146
Implementing SoA Particle Storage	146
SIMD-Accelerated Force Calculations	147
Fixed Time-Step Integration Loop	147
Performance Analysis and Optimization	148
Visualization Options	149
Summary	150
Chapter 24: Project 4 - A Multithreaded Job System	151
Design Requirements and Threat Model	151
Thread Pool Implementation	151
Work Queue and Scheduling	152
Thread-Local Storage and Context Management	153
Synchronization and Correctness Guarantees	153
Testing for Races and Deadlocks	154
Summary	155
Chapter 25: Project 5 - A TCP Networking Service	156
Service Requirements and Architecture	156
Socket Programming with core:net	156
Connection Handling and Event Loop	157
Handling Multiple Concurrent Connections	157

CONTENTS

Protocol Design and Message Framing	158
Resource Management and Graceful Shutdown	158
Security Considerations	159
Performance and Scaling	160
Summary	160
Chapter 26: Build Systems, Optimization and Tooling	161
The Odin Build System	161
Performance Optimization	162
Profiling Tools	163
Debugging	164
Production Deployment	166
Summary	167
Chapter 27: Testing, Debugging and Quality Assurance	168
Odin's Built-in Testing Framework	168
Property-Based Testing	169
Testing Patterns and Best Practices	169
Advanced Debugging Techniques	170
Static Analysis and Code Quality	171
Continuous Quality	172
Summary	173
Conclusion	174
What Makes Odin Unique	174
Skills You Gained	174
Next Steps	174
The Future of Odin	174
Final Thoughts	174
Appendix A: Language Quick Reference	176
Literals	176
Types	176
Procedures	176
Control Flow	176
Operators	176
Memory	177
Meta Operators	177
Attributes	177

Appendix B: Compiler and Tool Reference 178

 odin build 178

 odinfmt 178

 Odin Language Server (OLS) 178

 Testing 178

 Useful Commands 179

Glossary 180

References 181

Design, Build and Optimize Systems Software in Odin

Odin is a modern systems programming language built around data-oriented design, explicit resource management and predictable execution. This book teaches you how to use Odin to build real-world software: parsers, allocators, simulations, concurrent job systems, network services and performance-critical applications. You will learn not only how to write Odin code but also how the language connects to memory, caches, threads and operating-system interfaces. Every chapter contains complete, compilable examples, architectural reasoning and grounded comparisons with C, C++, Rust and Go. The book targets the Odin dev-2026-07 compiler release with updates through dev-2026-09 where explicitly noted. Whether you are coming from C, Rust, Go, or another systems language, this guide will help you design and ship software in Odin with confidence.

Introduction

In late 2025 you open a terminal on a machine that is supposed to process millions of physics updates per second and realize the existing C++ codebase has become nearly impossible to reason about. There are a dozen layers of template metaprogramming generating opaque error messages. Someone has hidden allocations behind smart pointers that themselves wrap shared state behind mutexes. When performance matters nobody can say where the CPU is actually spending its time. You have been told Rust could help but you are not sure you want to spend months learning ownership rules and fighting the borrow checker while the product ships without you. You have been told Zig might be the answer but its ecosystem is fragmenting and its design choices are not yet proven in large-scale deployments. Go has been suggested but the garbage collector pauses and the lack of fine-grained memory control make it unsuitable for the latency profile you need.

This is the gap Odin aims to fill.

Odin is a systems programming language created by Bill Hall, known in the community as Ginger Bill. It has been under continuous development since 2016 and reached its dev-2026-07 milestone in July 2026. Odin is not a research language and it is not trying to be everything. It is trying to be a tool that competent systems programmers can use to build software that is fast, correct and understandable. It makes explicit decisions: no hidden control flow, no garbage collection by default, no macros, no compile-time metaprogramming, no exceptions, no methods. Instead it offers what many low-level programmers have been asking for: a simple, predictable language with first-class support for data-oriented design, manual memory management with practical tooling and C-compatible interoperability.

This introduction explains what you will find in this book, who it is for and what you need to know before you begin.

What This Book Is

This book is a complete, practical guide to Odin for serious systems programming. It is organized into a logical progression that takes you from language

fundamentals through advanced topics to full-scale projects. Each chapter is self-contained in the sense that you can read it and come away with working knowledge of that topic, but chapters also build on each other so reading sequentially is strongly recommended.

The book has three distinct parts woven into a single narrative. The first part covers the language itself: syntax, types, procedures, control flow, memory management and polymorphism. The second part covers systems programming with Odin: interfacing with operating systems, handling I/O, threading, synchronization and performance optimization. The third part consists of five substantial projects that integrate everything you have learned into working, production-quality software.

Every code example in this book is complete, compilable and tested against a specific Odin compiler version. You can copy the code, run it, modify it and learn from it. Where compilation instructions, platform requirements, or expected output are non-obvious, they are provided explicitly.

What This Book Is Not

This book is not a reference manual. Odin has excellent official documentation at odin-lang.org and package documentation at pkg.odin-lang.org. This book does not try to duplicate those resources. Instead it uses them as a foundation and builds a narrative around practical understanding.

This book is not a language advocacy piece. Odin has trade-offs. It is not mature enough for every use case. Its ecosystem is small compared to C, C++, Rust, or Go. These limitations are discussed honestly and in detail. You will know when Odin is the right tool and when it is not.

This book contains no exercises, quizzes, or homework problems. The examples and projects are substantial enough that working through them is exercise enough. If you want practice, write the projects alongside the text and extend them.

Who This Book Is For

This book is for you if:

- You are a systems programmer who wants to learn Odin seriously and build production software in it.
- You have worked with C, C++, Rust, Go, or a similar low-level language and want to understand how Odin compares.
- You care about performance, memory layout and cache efficiency and want a language that supports data-oriented design as a first-class concern.
- You are a game developer, embedded engineer, database contributor, protocol implementer, or anyone who builds software where performance and predictability matter.
- You are skeptical of language hype and want a grounded, technical examination of what Odin can and cannot do.

This book is not for you if:

- You are looking for a gentle introduction to programming. You should already understand variables, functions, memory, pointers and basic computer architecture.
- You need a complete, authoritative reference for every Odin keyword and library function. The official documentation and source code are better suited for that purpose.
- You want to build web applications with a garbage-collected runtime and dynamic type system. Odin is not that language.

Prerequisites

To get the most from this book you should be comfortable with:

- Basic computer science concepts: types, procedures, data structures and algorithms.
- At least one compiled systems language, preferably C. Understanding C pointers, memory layout and compilation gives you a strong foundation for Odin.
- Familiarity with modern development practices: version control, build tools, testing and debugging.
- A working knowledge of at least one major operating system: Linux, macOS, or Windows.

- Basic understanding of computer architecture: CPU registers, instruction execution, cache hierarchies and virtual memory.

If you know Rust you will recognize many similarities and some deliberate divergences. If you know Go you will appreciate Odin's simpler control flow but need to adjust to manual memory management. If you know C you will find Odin much more ergonomic while keeping the same low-level access. If you know C++ you will likely appreciate Odin's resistance to feature creep.

How to Read This Book

Read the book from the beginning. The chapters are ordered to build a coherent mental model. Chapter 1 explains Odin's design philosophy. Chapter 2 gets your development environment working. Chapters 3 through 12 cover the language core. Chapters 13 through 20 cover advanced topics including data-oriented design, FFI, concurrency and performance. Chapters 21 through 25 are projects that integrate everything. Chapters 26 and 27 cover build systems and testing. The conclusion ties everything together.

You do not need to read every chapter cover to cover on the first pass. If you are already comfortable with basic syntax you can skim Chapters 3 through 6 and dive deeper as needed. The projects in Chapters 21 through 25 are designed so you can read and understand them even if you do not type every line.

Every code listing is meant to be understood, not just copied. Read the explanations carefully. They explain why code is written a particular way, what trade-offs were considered and what could go wrong.

Technical Conventions

This book uses Odin dev-2026-07 as its reference compiler version. Features added in later dev releases are noted explicitly when they appear. Some features discussed are part of an evolving language and may change in future releases. Where possible, the book focuses on stable, well-established aspects of the language.

Code examples follow Odin's style conventions. Identifiers use snake_case. Packages are imported using their canonical names. Where multiple approaches exist, the idiomatic choice is shown first.

Commands meant to be typed in a terminal are shown inline. Output from those commands is shown as block text. Comments in code explain non-obvious details.

All URLs and references to external resources are collected in the References section at the end of the book.

Acknowledging the Ecosystem

Odin is developed primarily by Bill Hall with contributions from a small community. The ecosystem is young. Libraries exist for graphics (OpenGL, Vulkan, Metal, Direct3D), physics, math, serialization and networking, but they are not as numerous or mature as those for C or Rust. This book focuses on Odin's core language and standard library rather than third-party packages, which change more rapidly.

The Odin compiler is written in C++ and uses LLVM as its backend. It is open source and hosted on GitHub. The language itself is pragmatic and evolves based on practical feedback rather than theoretical purity. This is both a strength and a risk: the language stays focused and usable, but it is not frozen and breaking changes do occasionally happen.

What You Will Know After This Book

After reading this book you will be able to:

- Write idiomatic Odin code using the full range of language features.
- Design data structures with memory layout, cache behavior and performance in mind.
- Manage memory explicitly with allocators, arenas and the context system.
- Build concurrent systems using threads, thread pools and synchronization primitives.
- Interface with C libraries, operating-system APIs and hardware.
- Profile, debug and optimize Odin programs.
- Architect and implement complete projects: parsers, allocators, simulations, job systems and network services.
- Make informed decisions about when to use Odin and when to choose another language.

You will not be an Odin expert in the sense of having memorized every detail of the standard library or compiler internals. But you will have a solid, practical foundation that lets you read Odin code, write Odin code and understand the trade-offs involved in doing so.

Now let us begin.

Chapter 1: Why Odin? The Design Philosophy

The State of Systems Programming in 2026

The landscape of systems programming languages in 2026 is crowded and fractured. At one end sits C, the language that has underpinned operating systems, databases, compilers and embedded firmware for fifty years. C is everywhere and C is simple enough that any competent engineer can read it after a few hours. But C's simplicity is also its curse. It gives you access to memory without telling you when you have fallen off the end of an array. It lets you cast pointers until your head spins and then wonder why the segmentation fault happened three functions deep. Buffer overflows, use-after-free bugs and integer overflows are not accidents in C. They are what you get when a language treats memory as a bag of bytes with no opinions about how you use it. The 2018 discovery of the Spectre and Meltdown vulnerabilities reminded everyone that unsafe memory access is not just a bug but a security risk that can survive for decades without anyone noticing [1].

At the other end sits Rust, the language that promises memory safety without garbage collection through a sophisticated ownership and borrowing system. Rust is genuinely impressive. Its borrow checker catches entire classes of bugs that C and C++ programs suffer from silently. Its standard library is excellent. Its ecosystem has grown to include operating system kernels, web browsers and database engines. But Rust is also complex. Learning to think in lifetimes and borrowing rules takes real time. Large Rust codebases can become exercises in satisfying the compiler rather than expressing intent. Template-like generic programming with traits and associated types can produce error messages that resemble cryptographic hashes. And while Rust has made enormous progress, its complexity is not going away. The language keeps growing and the learning curve keeps steepening.

Between these poles sit languages trying to find their own niches. Go chose simplicity and garbage collection over raw control and performance. It excels

at networking services, distributed systems and tooling but is a poor fit for latency-sensitive applications where allocation pauses are unacceptable. C++ is arguably the most capable systems language ever designed but its decades of feature accumulation have made it unwieldy. You can write C in C++ and functional programming in C++ and object-oriented programming in C++ and metaprogramming so aggressive it becomes unreadable. The problem is not that C++ is powerful. The problem is that its power has no guardrails.

Zig arrived with a promise to be a better C that is also simpler and more maintainable. It brings comptime execution, better array bounds checking and a sensible build system. Zig has a passionate following and its design decisions are interesting. But as of 2026 Zig is still finding its footing. Breaking changes happen. The ecosystem is growing but slowly. And the decision to have comptime execution built into the language has created some of the same complexity issues that Rust faces, just in a different domain.

This is the context in which Odin emerged. Odin is not trying to be the most powerful language. It is not trying to be the safest. It is trying to be the most pragmatic tool for systems programmers who know what they are doing and want a language that respects their intelligence.

What Odin Is and Is Not

Odin is a systems programming language created by Bill Hall, a game developer and programmer who has been building tools and engines for over twenty years. He began work on Odin in 2016 after growing frustrated with the limitations of C++ in game development and wanting a language that supported data-oriented design without sacrificing low-level control. The name Odin comes from Norse mythology but the design philosophy is thoroughly practical.

Odin is a compiled, statically typed, manually memory-managed language with C-compatible ABI. It compiles to native code via LLVM and targets Windows, Linux, macOS, FreeBSD, OpenBSD and WebAssembly. The current reference release used throughout this book is Odin dev-2026-07, released in July 2026 [2].

What Odin is:

- A low-level language that gives you direct access to memory, pointers and CPU features.

- A data-oriented language with built-in support for structure of arrays layouts, SIMD vectors and cache-friendly iteration.
- A language designed by and for practical engineers. Every feature exists to solve a real problem that Hall or the community encountered.
- A language that respects the programmer's time. No metaprogramming nightmares. No hidden allocations. No magic.

What Odin is not:

- A safe language. Odin trusts you to manage memory correctly. If you dereference a null pointer the program crashes. If you read past the end of an array the program may corrupt memory. There is no borrow checker and no runtime enforcement of ownership.
- A garbage-collected language. Odin has optional garbage collection support but it is not the default and not the idiomatic approach. You allocate and free memory explicitly.
- An object-oriented language. Odin has no classes, no inheritance, no virtual methods. It has structs, procedures and composition via the using statement.
- A scripting language or high-level language. Odin is not designed for rapid prototyping of web services or data analysis. It is designed for building software where performance, predictability and resource control matter.

Odin's tagline is The Data-Oriented Language for Sane Software Development. That tagline captures two essential design goals. First, data orientation is not an afterthought or an optimization you apply after writing your code. It is built into the language from the ground up. Second, sanity matters. A language can be powerful and still drive its users insane. Odin aims to avoid that fate.

Data-Oriented Design as First-Class Concern

Data-oriented design is a programming philosophy that emphasizes the layout and access patterns of data over the abstraction and encapsulation of objects. In traditional object-oriented programming you group related data and behavior together into classes and manipulate objects through methods. In data-oriented design you arrange data in memory so that the CPU can process it efficiently.

Consider a simple example. Suppose you are building a game engine that tracks thousands of particles, each with a position, velocity and lifetime. An object-oriented approach might define a `Particle` class or struct and store particles in a dynamic array of `Particle` objects:

```
1 struct Particle {
2     position: [3]f32
3     velocity: [3]f32
4     lifetime: f32
5 }
6 particles: [dynamic]Particle
```

This approach is convenient. You can write `update_particle(p)` and it feels natural. But it is terrible for performance. When you iterate through the particles to update their positions, the CPU loads each `Particle` from memory into a cache line. A cache line is typically sixty-four bytes. Each `Particle` takes up twenty-eight bytes, so each cache line holds only two particles. The third quarter of every cache line is wasted on other particles you have not loaded yet. If your cache is small or your particle count is large, you will suffer cache misses and your update loop will be limited by memory bandwidth rather than CPU speed.

A data-oriented approach stores each field in a separate contiguous array:

```
1 struct Particles {
2     positions: []f32      // [x0, y0, z0, x1, y1, z1, ...]
3     velocities: []f32     // [vx0, vy0, vz0, vx1, vy1, vz1, ...]
4     lifetimes: []f32     // [lt0, lt1, lt2, ...]
5     count: int
6 }
```

Now when you update positions, the CPU loads cache lines full of position data. No wasted space. No unnecessary loading of velocity or lifetime data that you do not need for this particular update. And because the data is contiguous, you can use SIMD instructions to update multiple particles in a single instruction.

Odin supports this pattern directly with its `#soa` modifier:

```
1 Particle :: struct {
2     position: [3]f32
3     velocity: [3]f32
4     lifetime: f32
5 }
6 particles: #soa[1000]Particle
7 // particles.position is now a contiguous array of all positions
8 // particles.velocity is now a contiguous array of all velocities
9 // particles.lifetime is now a contiguous array of all lifetimes
```

This is not a library feature or a convention you have to remember. It is part of the language. You can also use native SIMD types, matrix operations, quaternion arithmetic and array swizzling directly in Odin without importing external libraries. These features are not bolted on as afterthoughts. They are core parts of the language design.

This commitment to data orientation runs deeper than syntax. Odin's type system encourages you to think about memory layout. Struct fields keep their declaration order, with C-style alignment padding. You can control alignment explicitly. The language provides built-in procedures for measuring sizes, offsets and alignments. The compiler tells you when your code will likely cause cache inefficiencies. This is a language designed by someone who has spent years optimizing game loops and knows that how data sits in memory matters more than almost anything else.

The Pragmatist's Manifesto: Simplicity Over Power

Odin's design is guided by pragmatism. Bill Hall has written explicitly about his approach [3]. He does not design features because they are theoretically interesting. He designs features because they solve real problems he or other Odin users have encountered. If a feature adds more complexity than value, it gets rejected.

This principle has led to some surprising decisions. The most notable is the rejection of compile-time metaprogramming. Many modern systems languages have some form of compile-time execution or macro system: C has the preprocessor, C++ has templates, Rust has macros, Zig has comptime. When Hall first designed Odin he considered adding a similar feature. He then spent time prototyping it and concluded that it was more trouble than it was worth.

In a July 2025 article titled *If Odin Had Macros*, Hall explains his reasoning [4]. Compile-time metaprogramming sounds powerful but it creates real problems for programmers. Macros obscure the actual code being executed. Compile-time execution can create confusing error messages that are difficult to debug. The power comes at the cost of predictability. Hall concluded that every problem he thought required metaprogramming in Odin could be solved with simpler language features like parametric polymorphism, where clauses, or the `when` statement. So Odin has no macros. No comptime. No AST-level code generation. This is a deliberate design choice and it makes Odin code straightforward to read and understand.

Similarly, Odin has no exceptions. Exceptions are powerful for error handling but they introduce hidden control flow. A function call might throw an exception that propagates through dozens of call frames before being caught. You cannot see that control flow by reading the code. You have to know which functions throw and which catch. In a systems language where performance and predictability matter, exceptions are a liability. Odin handles errors through multiple return values and explicit checks. This is not new. Go does it. C does it. But Odin adds syntactic support through the `or_return` pattern that makes error handling concise without sacrificing clarity.

Odin also has no methods. Procedures always take their first argument explicitly rather than using receiver syntax like Go's `func (p *Person) Name()` string. This might seem like a minor point but it is not. Methods encourage object-oriented thinking and implicit receiver semantics that can hide what is actually happening. Odin prefers explicit procedure calls. It provides the `using` statement for composition, which is cleaner and more flexible than inheritance.

The calling convention is another example of pragmatic design. Odin has its own calling convention called `odin` that passes the implicit context pointer in a hidden parameter. The context carries allocators, logging configuration, random number generators and other thread-local state. This eliminates the need for global state while keeping function signatures clean. When interfacing with C code you can use the standard C calling convention. When performance matters and you want to avoid the context overhead you can mark a procedure as contextless. The language gives you choices without forcing one pattern on everything.

Comparison Landscape: C, C++, Rust, Go, Zig

Understanding where Odin fits requires comparing it to the languages that define the systems programming landscape. Each comparison highlights different design choices and trade-offs.

Odin vs C

C is Odin's spiritual ancestor. Odin keeps C's fundamental execution model: manual memory management, direct pointer access, C-compatible ABI, no hidden control flow. But Odin fixes almost everything that makes C painful to use in 2026.

Where C has separate compilation units with header files and forward declarations, Odin has a unified package system where all files in a directory form a single package. There are no header files. There is no `#include` cascade. You import packages by name and the compiler handles dependencies.

Where C has macros for code generation and conditional compilation, Odin has when statements for compile-time branching and parametric polymorphism for generic code. No more worrying about macro expansion order or accidental token concatenation.

Where C has `sizeof` but no way to query alignment or field offsets without compiler-specific extensions, Odin has `size_of`, `align_of` and `offset_of` as built-in procedures. The compiler also provides runtime type information through `typeid` and the `reflect` package.

Where C has a minimal standard library, Odin includes a comprehensive core library with string handling, file I/O, networking, threading, memory allocators, math operations, serialization and more. None of this is optional. It ships with the compiler.

The trade-off is that Odin code does not compile with a C compiler. You need the Odin toolchain. But for new projects that is rarely a limitation.

Odin vs C++

C++ is Odin's antithesis in many ways. C++ has grown for forty years and contains almost every programming paradigm. Odin is deliberately small and focused.

C++ has classes, inheritance, virtual functions, templates, constexpr, coroutines, concepts, modules and a standard library that fills multiple books. Odin has structs, procedures, parametric polymorphism and a standard library that is comprehensive but concise.

C++ manages memory through a combination of raw pointers, stack allocation, RAII destructors, smart pointers and sometimes garbage collectors. Odin manages memory explicitly through allocators and the context system. There are no destructors but there is defer, which provides similar scoped cleanup behavior without the object-oriented baggage.

C++ templates are Turing-complete. You can write a sort algorithm in the type system. Odin's parametric polymorphism is simpler. It generates monomorphized code for each type used but does not allow arbitrary computation at compile time. The result is code that is easier to read, easier to debug and faster to compile.

If you are coming from C++ you will likely appreciate Odin's resistance to feature creep. You will not find yourself wondering which of the twelve ways to express a generic operation is the idiomatic choice. Odin has one way and it works.

Odin vs Rust

Rust and Odin share many goals: memory safety (or at least explicit memory management), performance and a modern developer experience. But they reach for those goals in very different ways.

Rust enforces memory safety through compile-time borrow checking. Odin trusts the programmer to manage memory correctly. Rust catches use-after-free and data races at compile time. Odin does not. Rust has a steep learning curve but produces provably correct memory usage. Odin has a shallow learning curve but puts the burden on the developer.

Rust has traits for generic programming. Odin has parametric polymorphism with where clauses. Both generate monomorphized code. Rust's trait system is more expressive but also more complex.

Rust has lifetimes to track how long references are valid. Odin has no lifetime annotations. You reason about lifetimes manually.

Rust's standard library and ecosystem are mature. Odin's are younger but growing.

The choice between Rust and Odin often comes down to your risk tolerance. If you need compile-time guarantees and can accept the complexity, Rust is excellent. If you want simplicity and are confident in your ability to write correct memory-managed code, Odin is attractive.

Odin vs Go

Go and Odin share a love of simplicity and explicit control flow. Both reject exceptions and complex type systems. But they diverge on memory management and performance.

Go has garbage collection. Odin does not. Go's goroutines are lightweight and scheduled by the runtime. Odin's threads map directly to operating-system threads. Go's channels provide structured concurrency. Odin provides mutexes, semaphores and atomics.

Go is designed for network services and distributed systems. Odin is designed for low-level systems programming and game development. Go hides complexity behind its runtime. Odin exposes it.

If you build web services, Go is probably a better fit. If you build game engines, databases, or anything where allocation latency matters, Odin gives you more control.

Odin vs Zig

Zig and Odin are perhaps the closest competitors. Both aim to be better C. Both emphasize simplicity, manual memory management and C interoperability.

Zig has comptime execution built into the language. Odin rejected that approach in favor of separate build-time tools and simpler language features.

Zig uses a unique build system written in Zig itself. Odin has a built-in build command that is simpler but less flexible.

Zig has a single standard library that is growing. Odin has a core library that is already comprehensive.

Zig's error handling uses error sets and try. Odin uses multiple return values with `or_return`.

Both are young languages with small ecosystems. The choice often comes down to personal preference: do you want comptime power and a more unique design (Zig) or straightforward pragmatism and data-oriented features (Odin)?

The Odin Promise

Odin's promise is simple: it will be a language that competent systems programmers enjoy using. It will not hide complexity behind layers of abstractions. It will not force you to learn a new paradigm to do your job. It will give you direct control over memory and CPU resources while providing modern conveniences that make life easier.

The language is not perfect. It has rough edges. Some design decisions are controversial. The ecosystem is small. Breaking changes do happen. But Odin is also focused, honest about its goals and built by someone who actually uses it every day.

If that sounds appealing to you, the rest of this book will show you how to use Odin to build real software. Let us move from philosophy to practice.

Chapter 2: Setting Up the Development Environment

Before writing any substantial code you need a working Odin development environment. This chapter walks you through installing the Odin compiler, verifying the installation, building your first program and setting up editor tooling. We will use Odin dev-2026-07 as the reference version throughout this book.

Installing the Odin Compiler

Odin provides official binary releases for Windows, Linux and macOS. The releases are available from the Odin GitHub repository at <https://github.com/odin-lang/Odin/releases> [2]. Odin also distributes through various package managers for convenience.

Official Binary Releases

To install Odin from official releases, follow these steps:

1. Visit <https://github.com/odin-lang/Odin/releases>
2. Navigate to the dev-2026-07 release (or later, but be aware of breaking changes)
3. Download the appropriate archive for your platform:
 - Windows: `odin-dev-2026-07-win64.zip`
 - Linux x86_64: `odin-dev-2026-07-linux.tar.gz`
 - Linux arm64: `odin-dev-2026-07-linux-arm64.tar.gz`
 - macOS Intel: `odin-dev-2026-07-macos.zip`
 - macOS Apple Silicon: `odin-dev-2026-07-macos-arm64.zip`
4. Extract the archive to a permanent location. For example:
 - Windows: `C:\odin`

- Linux/macOS: `/usr/local/odin`

5. Add the Odin bin directory to your system PATH:

- Windows: Add `C:\odin\bin` to PATH
- Linux/macOS: Add `/usr/local/odin/bin` to PATH (edit `~/.bashrc`, `~/.zshrc`, or similar)

6. Verify installation by running `odin` version in your terminal. You should see output similar to:

```
1    Odin dev-2026-07
```

Platform-Specific Requirements

Odin relies on external C toolchains for linking. Make sure you have the appropriate compiler and SDK installed:

- **Windows:** Visual Studio Build Tools with the Desktop development with C++ workload. This provides the MSVC compiler and Windows SDK. If you do not have Visual Studio, download it from <https://visualstudio.microsoft.com/> and select the C++ desktop development component during installation [5].
- **Linux:** GCC or Clang with development headers. On Ubuntu or Debian, run:

```
1    sudo apt-get install build-essential
```

On Fedora:

```
1    sudo dnf install gcc gcc-c++ make
```

Odin also works with `lld` as a linker for faster builds. Install it with your package manager if desired.

- **macOS:** Xcode Command Line Tools. Install them by running:

```
1  xcode-select --install
```

On Apple Silicon Macs, Odin builds and runs natively on ARM64.

Package Manager Installation

Odin is available through several package managers. These are convenient but may lag behind official releases.

- **Homebrew (macOS/Linux):**

```
1  brew install odin
```

- **Chocolatey (Windows):**

```
1  choco install odin
```

- **Arch Linux:**

```
1  sudo pacman -S odin
```

- **Nix:**

```
1  nix-env -iA nixpkgs.odin
```

Check the official documentation for the most up-to-date package manager instructions [5].

Building from Source

Building Odin from source is straightforward and often desirable if you want the absolute latest changes or need to debug the compiler itself. The Odin compiler is written in C++ and uses LLVM as its backend.

Prerequisites:

- CMake 3.18 or later
- C++17 compatible compiler (GCC 9+, Clang 10+, MSVC 2019+)
- LLVM 15+ (LLVM 14 support was dropped in dev-2026-07) [2]

- libtommath (included in the Odin repository)
- Git

Steps:

1. Clone the repository:

```
1  git clone https://github.com/odin-lang/Odin.git
2  cd Odin
```

2. Check out the desired release tag:

```
1  git checkout dev-2026-07
```

3. Run the build script for your platform:

- Windows: build.bat
- Linux/macOS: ./build_odin.sh

The build script handles fetching dependencies, configuring CMake and compiling. It produces the odin binary in the current directory. On a modern machine, building from source typically takes five to fifteen minutes depending on CPU cores and LLVM size.

To build a release version with optimizations, use the release mode:

- Windows: build.bat release
- Linux/macOS: ./build_odin.sh release

The release build is slower to compile but produces a faster Odin compiler.

Your First Program: Hellope World

Odin's traditional first program is not Hello World. It is Hellope World, a nod to the Odin language's tagline about programming with joy. Create a new directory for your first project:

```
1 mkdir hellope
2 cd hellope
```

Create a file named `main.odin` with the following content:

```
1 package main
2
3 import "core:fmt"
4
5 main :: proc() {
6     fmt.println("Hellope World!")
7 }
```

This is a complete Odin program. Let us break it down:

- `package main`: Every Odin file begins with a package declaration. The package name must be unique across all packages in a build and serves as a prefix for symbol names to ensure consistent linking. The name `main` is conventional for executable programs [6].
- `import "core:fmt"`: This imports the `fmt` package from Odin's core library. The `core` prefix indicates that this is an official package that ships with the Odin compiler. The `fmt` package provides formatted input and output functions similar to C's `printf` family.
- `main :: proc()`: This declares the main procedure, which is the entry point of the program. The `::` operator is used for constant and procedure declarations. `proc` indicates that `main` is a procedure (Odin's term for function). The empty parentheses mean it takes no parameters and returns no values.
- `fmt.println("Hellope World!")`: This calls the `println` procedure from the `fmt` package, which prints a string followed by a newline.

To run the program, use the `odin run` command from the directory containing `main.odin`:

```
1 odin run .
```

The `.` argument tells Odin to compile all `.odin` files in the current directory as a package. Odin compiles the code, links it and runs the resulting executable. You should see:

```
1 Hellope World!
```

To build without running, use `odin build`:

```
1 odin build .
```

This produces an executable named `hellope.exe` on Windows or `hellope` on Linux and macOS, named after the directory.

To run with a custom output name:

```
1 odin build . -out:my_hellope
```

This produces `my_hellope.exe` or `my_hellope` instead.

The `odin` Command: Build, Run, Test, Check

The `odin` command is the primary interface to the Odin toolchain. It handles compilation, linking, testing and more. Understanding its modes and flags is essential for effective development.

`odin run`

The `run` command compiles and immediately executes your program. This is the most convenient way to iterate on small programs and examples.

```
1 odin run .  
2 odin run my_project
```

By default, `run` uses minimal optimization and includes debug information. This makes compilation fast and debugging easy. For larger projects, consider using `build` for release versions and `run` only during development.

odin build

The build command compiles and links without running. It supports many options:

- `-out:name`: Specify output file name
- `-debug`: Include debug symbols (enabled by default for run)
- `-o:level`: Set optimization level (none, minimal, moderate, aggressive)
- `-no-bounds-check`: Disable array bounds checking for release builds
- `-target:platform`: Cross-compile for a different platform
- `-microarch:native`: Optimize for the host CPU's instruction set
- `-lto:thin`: Enable link-time optimization

Examples:

```
1 # Debug build with debug symbols
2 odin build . -debug
3
4 # Release build with aggressive optimization and no bounds checking
5 odin build . -o:aggressive -no-bounds-check
6
7 # Build for Linux ARM64 from a Linux x86_64 machine
8 odin build . -target:linux_arm64
9
10 # Build with link-time optimization
11 odin build . -lto:thin
```

The default optimization level for build is minimal, which provides reasonable performance without significantly slowing compilation. For production binaries, use moderate or aggressive [7].

odin test

The test command compiles and runs test procedures marked with the `@(test)` attribute. We will cover testing in detail in Chapter 27. For now, know that the basic usage is:

```
1 odin test .  
2 odin test . -all-packages
```

Odin's test runner is multithreaded by default and can track memory allocations. Configuration is done through compile-time defines like `ODIN_TEST_THREADS` and `ODIN_TEST_TRACK_MEMORY`.

odin check

The `check` command parses and type-checks your code without generating an executable. This is faster than `build` and useful for CI pipelines, linting and IDE integration.

```
1 odin check .  
2 odin check . -vet
```

The `-vet` flag enables additional static analysis checks for common errors and style issues.

odin bundle

The `bundle` command packages your project for distribution in a platform-specific format. On Windows it produces a folder structure suitable for installation. On Linux it follows the Filesystem Hierarchy Standard. On macOS it can produce a `.app` bundle.

```
1 odin bundle my_project
```

odin version

Displays the Odin version and build information:

```
1 odin version
```

Use this to verify your installation and to report when filing bugs.

Editor and Language Server Setup with OLS

Writing Odin code is pleasant in any decent text editor but it is significantly better with language server support. The Odin Language Server (OLS), developed by Daniel Gavin, provides syntax highlighting, autocompletion, go-to-definition, type information, diagnostics and more [8].

Installing OLS

OLS is available from its GitHub repository at <https://github.com/DanielGavin/ols>. Binary releases are provided for Windows, Linux and macOS.

Download the latest release and extract it to a location on your PATH. On most systems:

```
1 mkdir -p ~/.local/bin
2 # Extract ols binary to ~/.local/bin
```

Then ensure `~/.local/bin` is in your PATH.

Alternatively, OLS can be built from source using the Odin compiler:

```
1 git clone https://github.com/DanielGavin/ols.git
2 cd ols
3 odin build .
```

This produces an `ols` binary in the current directory.

VS Code Setup

For Visual Studio Code, install the Odin extension from the marketplace:

1. Open VS Code and go to Extensions
2. Search for Odin
3. Install the Odin language extension (typically by Ginger Bill or a community maintainer)
4. In VS Code settings, set the path to OLS:

```
1  {
2    "odin.ols.path": "/path/to/ols"
3  }
```

Neovim Setup

For Neovim users with nvim-lspconfig:

```
1  require('lspconfig').ols.setup({
2    cmd = { "ols" },
3    filetypes = { "odin" },
4    root_dir = function(fname)
5      return vim.fs.dirname(vim.fs.find({ ".git" }, { upward = true, path =
6        ↪ fname })[1])
7    end,
8  })
```

Other Editors

OLS implements the Language Server Protocol, so it works with any editor that supports LSP. Sublime Text, Emacs, Vim, Zed and Helix all have Odin support through OLS.

OLS Configuration

OLS reads configuration from an `ols.json` file in your project root. A minimal configuration:

```

1  {
2      "odin_command": "odin",
3      "odin_root_override": "",
4      "profile": "default",
5      "profiles": [
6          {
7              "name": "default",
8              "checker_path": [ "." ],
9              "defines": {
10                 "ODIN_DEBUG": "true"
11             }
12         }
13     ]
14 }

```

The `checker_path` tells OLS which directories to check. The `defines` section sets compile-time constants that OLS uses for type checking. Multiple profiles allow different configurations for different platforms or build modes [8].

Project Structure and Collections

Understanding Odin's project structure is crucial because it is fundamentally different from many other languages.

Packages Are Directories

In Odin, a package is simply a directory containing `.odin` files. All files in that directory must declare the same package name. There are no header files. There is no separate declaration and definition split. The compiler treats all files in a directory as a single compilation unit.

Example directory structure:

```

1  my_project/
2      main.odin          // package main
3      utils.odin         // package main (same package, same directory)
4      math/
5          vector.odin     // package math
6          matrix.odin     // package math

```

When you run `odin build .` in `my_project`, the compiler compiles `main.odin` and `utils.odin` together as the main package. If `main.odin` needs vector operations, it imports the `vector_math` package:

```

1 package main
2
3 import vm "./vector_math"
4 import "core:fmt"
5
6 main :: proc() {
7     v := vm.Vector3{1.0, 2.0, 3.0}
8     fmt.Println(v)
9 }

```

Import Names and Aliases

Import names are chosen at the import site, not based on the directory name. The identifier before the path string is the local alias for the package and this is the only aliasing mechanism Odin has - there is no trailing `as` clause. This gives you flexibility and avoids naming conflicts:

```

1 import strings "./my_strings" // import name is strings
2 import fmt "core:fmt"        // import name is fmt
3 import vm  "./vector_math"   // alias the vector_math package as vm

```

Note that package names must be unique across the whole build. Naming your own package `math` would collide with `core:math` as soon as both appear in the build graph, so pick a distinctive name such as `vector_math`.

The import name is used as a prefix for all symbols from that package. This is different from Go, where the last element of the import path becomes the package name.

Collections

Odin organizes packages into three collections:

- **base:** Built-in packages that are always available. These include builtin (predeclared identifiers like `int`, `bool`, `len`, `size_of`), intrinsics (compiler intrinsics) and runtime (runtime support code). You cannot import these directly with the import statement. Their symbols are always in scope.
- **core:** Official packages that ship with the Odin compiler and are considered stable. These include `fmt`, `os`, `io`, `mem`, `thread`, `sync`, `net`, `slice`, `strings`, `reflect` and many others. Import with the core prefix: `import "core:fmt"`.

- **vendor:** Official bindings for third-party libraries. These include OpenGL, Vulkan, Direct3D, Metal, SDL and other common libraries used in game development and graphics programming. Import with the vendor prefix: `import "vendor:gl" [9]`.

User-defined packages are imported using relative or absolute paths. Odin does not have a global package registry or official package manager. You manage dependencies by including them in your project directory or through your own vendoring system.

Why No Package Manager?

Odin's creator has explicitly stated that Odin will never have an official package manager [6]. The reasoning is that package management is a complex problem with many legitimate approaches and choosing one would lock Odin into a particular ecosystem model. Instead, Odin provides the tools for you to manage dependencies your own way: vendoring, submodules, custom scripts, or third-party tools like draupnir.

This is a controversial design decision. It means Odin developers must solve dependency management themselves. But it also means Odin's core stays small and focused on language features rather than ecosystem infrastructure.

Verifying Your Setup

Before moving on to Chapter 3, verify that your development environment is working correctly. Create a small test project and run it:

1. Create a directory: `mkdir test_odin && cd test_odin`
2. Create `main.odin`:

```
1  package main
2
3  import "core:fmt"
4
5  main :: proc() {
6      fmt.println("Odin version:", ODIN_VERSION)
7      fmt.println("Target OS:", ODIN_OS)
8      fmt.println("Target Arch:", ODIN_ARCH)
9      fmt.println("Debug mode:", ODIN_DEBUG)
10 }
```

3. Run it: `odin run .`

Expected output should show your Odin version, operating system, architecture and whether you are in debug mode. If this works, your environment is ready.

This chapter has given you a working Odin development environment, basic familiarity with the `odin` command and an understanding of how Odin organizes code into packages. The next chapter dives into language fundamentals: declarations, types and values.

Chapter 3: Language Fundamentals:

Declarations, Types and Values

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Statements, Expressions and Automatic Semicolon Insertion

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Variables: Declaration, Initialization and Zeroing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Zero Initialization

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Type Inference

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Multiple Declaration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Scope and Shadowing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Mutation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Constants and Compile-Time Expressions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Compile-Time Limits

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

No Runtime Address

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Primitive Types: Integers, Floats, Booleans, Runes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Integer Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Floating-Point Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Boolean Type

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Rune Type

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Endianness Variants

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Byte Type

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Distinct Types vs Aliases: Type Identity Matters

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Type Aliases

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Distinct Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Built-In Distinct Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Aggregate Types Are Always Distinct

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Literals: Numbers, Strings and the Underscore Convention

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Numeric Literals

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

String Literals

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Array and Struct Literals

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 4: Procedures: Functions with Intent

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Declaring and Calling Procedures

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Parameters

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Named Parameters

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

No Default Parameters

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Parameters, Named Results and Multiple Return Values

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Basic Multiple Returns

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Named Results and Documentation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Ignoring Return Values

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Returning Multiple Values in Expressions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The Relationship to Error Handling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Calling Conventions: `odin`, `c`, `contextless` and Beyond

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The `odin` Calling Convention

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The `c` Calling Convention

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The `contextless` Calling Convention

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Other Calling Conventions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Specifying the Default Calling Convention for Foreign Blocks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Variadic Procedures and Type Safety

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Basic Variadics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Passing Arrays and Slices

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Any Variadics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Variadics Are Type-Safe

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Defer and Scope-Guaranteed Cleanup

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Basic Defer

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Defer for Resource Cleanup

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Deferred Blocks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Defer Semantics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Defer Has a Cost

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Tail Calls and Recursion

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Natural Tail Calls

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Non-Tail Calls

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

The #must_tail Directive

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Recursion vs Iteration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 5: Control Flow Without Hidden Complexity

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Range-Based For Loops

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Range Iteration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Iterating Over Collections

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Traditional C-Style For Loops

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Breaking and Continuing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Labeled Loops

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Unrolling Loops

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

If Statements and Ternary Expressions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Basic If

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

If-Else

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chained If-Else

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

If with Initialization

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

No Ternary Expression

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

No Implicit Truthiness

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Switch: Exhaustive, No Fallthrough, Pattern-Match-Free

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Basic Switch

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Multiple Values in a Case

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Exhaustive Matching for Enums

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

#partial switch

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Switch with Conditions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

No Pattern Matching

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

The or Family: or_return, or_else, or_break, or_continue

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

or_return

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

or_else

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

or_break and or_continue

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Combining the or Operators

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

The When Statement: Compile-Time Branching

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Platform-Specific Code

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Feature Flags

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Type-Specific Code

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

When vs If

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Why No Goto and No Exceptions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

No Goto

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

No Exceptions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 6: Composite Types: Structs, Unions, Enums and Bit Sets

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Structs: Layout, Alignment, Padding and #packed

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Declaring Structs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Creating and Using Structs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Nested Structs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Struct Layout and Padding

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

#packed: Eliminate Padding

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

#align: Specify Alignment

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Anonymous Structs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Unions and #raw_union: Tagged vs Untagged

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Basic Unions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Odin Unions vs C Unions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

#raw_union: C-Compatible Unions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Tagged Unions (Discriminated Unions)

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Enums and Enumerated Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Basic Enums

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Backing Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Enum Usage

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Enumerated Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Bit Sets: Type-Safe Flag Management

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Declaring Bit Sets

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Using Bit Sets

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Bit Sets vs Enums

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Fields, Tags and the Using Statement

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Field Tags

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The Using Statement

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Using with Unions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Aggregate Initialization and Partial Initialization

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Struct Initialization

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Array Initialization

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Enumerated Array Initialization

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Zero Initialization

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 7: Memory Layout: From Bytes to Caches

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

How the Compiler Lays Out Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Alignment Requirements

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Declaration-Order Layout

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Why Odin Does This

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Pointer and String Sizes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Alignment, Packing and ABI Requirements

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Declaration Order and C Compatibility

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

The #packed Tag

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

The #align Tag

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

ABI and Calling Conventions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Measuring Size and Offset with `sizeof` and `alignof`

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

`size_of`

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

`align_of`

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

`offset_of`

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

`container_of`

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Cache Lines, Prefetching and Data Locality

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

The Memory Hierarchy

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Cache Lines

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Spatial Locality

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Temporal Locality

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

False Sharing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Prefetching

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Struct Layout Optimization Techniques

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Order Fields by Access Frequency

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Group Hot and Cold Fields

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Minimize Struct Sizes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Use Smaller Types When Possible

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Matching C Layouts: Foreign Interop and Hardware Formats

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Foreign Structs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Binary File Formats

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Hardware Registers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 8: Arrays, Slices and Dynamic Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Fixed Arrays: Stack-Bound and Sized

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Declaring Fixed Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Fixed Arrays Are Values

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Fixed Arrays on the Stack

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Accessing Elements

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Length and Capacity

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Arrays of Structs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Slices: Pointers with Length

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Slice Structure

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Creating Slices

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Slices Share Data

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Slices Have Length, Not Capacity

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Empty and Nil Slices

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Slices and Procedures

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Dynamic Arrays: Growth, Allocation and Ownership

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Declaring Dynamic Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Dynamic Array Structure

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Growing Dynamic Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Preallocating Capacity

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Deleting Dynamic Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Dynamic Arrays vs Slices

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Slice Operations: Append, Delete and Copy

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

append

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

delete

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

copy

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

pop

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

resize and reserve

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Array Arithmetic and Component-Wise Operations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Vector Arithmetic

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Scalar Operations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Swizzling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

When to Use Each Array Type

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Use Fixed Arrays When:

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Use Slices When:

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Use Dynamic Arrays When:

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Anti-Patterns to Avoid:

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 9: Maps and String Handling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

The Map Type: Hash Table Semantics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Declaring Maps

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Creating Maps

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Map Operations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Map Size and Capacity

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Map Semantics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Map Performance

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Map Keys: What Is Allowed

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Valid Key Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Invalid Key Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Strings as Pointer Plus Length

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

String Literals

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

String Immutability

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

String Operations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

String to Slice and Back

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

UTF-8, Runes and Character Encoding

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Iterating by Bytes vs Runes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Rune Type

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

String Indexing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The core:strings Package

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

C Strings and Interop with External APIs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Converting Odin Strings to C Strings

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Converting C Strings to Odin Strings

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The cstring Type

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Performance Considerations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

String Interning and Pooling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 10: Pointers, Memory Addresses and Ownership

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Single Pointers: Λ T Semantics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Declaring Pointers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Dereferencing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Pointers and nil

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Pointers to Structs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

No Arrow Operator

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Multi-Pointers: [^]T for Typed Untyped-Length Access

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Declaring Multi-Pointers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Multi-Pointer Indexing and Slicing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Converting Between Pointers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Multi-Pointers and Slices

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Rawptr and Type Erasure

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Declaring Rawptr

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Casting Rawptr

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Rawptr and Memory Operations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Type Erasure with Rawptr

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Pointer Arithmetic and Unsafe Access

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Indexing Through Multi-Pointers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Walking Through Memory

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Undefined Behavior

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Memory Ownership: The Programmer's Responsibility

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Ownership Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Caller Allocates, Caller Frees

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Callee Allocates, Caller Frees

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Owned vs Borrowed

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Returning Pointers to Stack Memory

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Use-After-Free

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Common Pointer Mistakes and Corruption Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Dangling Slices

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Double Free

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Memory Leaks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Off-by-One Errors

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Type Confusion

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 11: The Context System and Memory Allocators

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

What the Context Is and How It Flows

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

The Context Struct

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

How the Context Flows

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Context Per Thread

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Context Fields: Allocators, Logger, RNG and More

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

allocator and temp_allocator

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

assertion_failure_proc

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

logger

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

random_generator

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

user_ptr and user_index

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

new, make, free and delete

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

new

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

make

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

free

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

delete

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Allocation Without Zeroing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The Temporary Allocator: Arena-Style Allocation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Why Arena Allocation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Using the Temp Allocator

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Manual Arena Allocators

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

When to Use Arena Allocation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Tracking Allocator: Debugging Memory Issues

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Using the Tracking Allocator

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Odin Test Runner Integration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Custom Allocators and Context Overrides

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The Allocator Interface

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Pool Allocator Example

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Context Override Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 12: Parametric Polymorphism: Generic Procedures and Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Explicit vs Implicit Parametric Polymorphism

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Implicit Parametric Polymorphism

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Explicit Parametric Polymorphism

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Mixing Explicit and Implicit

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Where Clauses and Type Constraints

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Basic Where Clauses

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Available Type Constraints

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Size and Alignment Constraints

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Custom Constraints with Ininsics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Parametric Structs: Generic Data Structures

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Basic Parametric Structs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Generic Containers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Implicit Type Parameters for Structs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Type Inference and Its Limits

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

When Inference Works

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

When Inference Fails

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Inference with Polymorphic Procedures as Arguments

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Type-Specific Code Paths (No Overloading)

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Distinct Names for Concrete Versions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Branching on T Inside a Polymorphic Procedure

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

What Odin Does Not Have

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Comparison with Rust Traits and C++ Templates

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Odin vs Rust Traits

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Odin vs C++ Templates

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Odin vs Go Generics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 13: Data-Oriented Design with Odin

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

What Data-Oriented Design Means

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

AoS vs SoA vs AoSoA: Memory Layouts for Performance

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Array of Structs (AoS)

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Structure of Arrays (SoA)

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Array of Structures of Arrays (AoSoA)

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The #soa Modifier: Odin's Struct of Arrays

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Declaring SOA Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Accessing SOA Fields

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

SOA and Slices

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

SOA Limitations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

SIMD Vectors: Native Support and Swizzling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

SIMD Vector Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Using core:simd

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

SIMD and SOA Together

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Swizzling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Matrix and Quaternion Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Matrices

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Quaternions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Designing Cache-Friendly Data Structures

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Keep Hot Data Together

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Minimize Pointer Indirection

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Use Appropriate Granularity

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Profile Before Optimizing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Chapter 14: Error Handling: Explicit by Design

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Why No Exceptions: Design Rationale

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Hidden Control Flow

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Resource Management Complexity

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Performance and Predictability

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

No Exceptions in Odin

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Multiple Return Values for Success and Error

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The (value, ok) Pattern

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The (value, error) Pattern

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Choosing Between Patterns

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The or_return Pattern

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Basic or_return

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

or_return Through Layers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Readability Trade-offs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Readability Trade-offs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Custom Error Types and Discriminated Unions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Error Enums

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Discriminated Unions for Detailed Errors

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Result Type

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Error Propagation Strategies

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Early Return

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Error Accumulation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Panic for Truly Unrecoverable Errors

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Comparison with Go and Rust Approaches

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Odin vs Go

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Odin vs Rust

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 15: Packages and the Module System

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Packages: Directories as Units of Compilation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Package Declaration

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

All Files in a Directory Share One Package

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Why This Design?

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Import Names and Aliases

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Basic Imports

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Import Aliases

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Import Paths

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

No Implicit Imports

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Visibility: @private, #+private and Export Control

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

@(private) Attribute

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

#+private File Directive

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Visibility Rules

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Why No Private by Default?

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Core, Vendor and User Collections

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

base: Built-In Packages

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

core: Official Packages

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

vendor: Third-Party Bindings

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

User Packages

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Foreign Import and Foreign Blocks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Foreign Import

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Foreign Blocks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Attributes for Foreign Symbols

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Calling Conventions in Foreign Blocks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Type Mapping

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Structuring Large Projects

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Project Layout

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Package Dependencies

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Shared Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Implementation Details

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 16: Foreign Function Interface: Calling C and the OS

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Foreign Import: Linking to Libraries

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Basic Foreign Import

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Importing Custom Libraries

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Platform-Specific Imports

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Foreign Blocks: Declaring External Symbols

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Basic Foreign Block

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Symbol Name Mapping

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Declaring Variables

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Procedural Types in Foreign Blocks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Type Mapping: Odin Types to C Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The core:c Package

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Why Not Use Odin's int?

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Struct Mapping

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Union Mapping

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Pointer Mapping

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Enum Mapping

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Array Mapping

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Calling Conventions and ABI Details

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The “c” Convention

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Other Conventions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Variadic Functions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Struct Passing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

String Interop: string vs cstring

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Odin string

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

C cstring

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Converting Odin string to cstring

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Converting cstring to Odin string

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Common Mistakes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Working with System Libraries: Windows, POSIX, macOS

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

POSIX Systems (Linux, macOS, FreeBSD)

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Windows

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

macOS

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Cross-Platform Code

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 17: File I/O, Networking and the Operating System

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

The New core:os v2: Design and API

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Key Design Principles

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Importing core:os

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Standard Streams

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

File Operations: Open, Read, Write, Seek

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Opening Files

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Reading Files

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Reading All Content

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Writing Files

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Creating Files

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Seeking

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Closing Files

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

File Metadata

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Paths, Directories and Environment

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Path Operations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Directory Operations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Environment Variables

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

User Directories

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Networking with core:net: Sockets and DNS

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

TCP Connections

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

UDP Sockets

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

DNS Resolution

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Socket Options

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Non-Blocking I/O with core:nbio

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Event Loop Basics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Non-Blocking Sockets

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Use Cases for Non-Blocking I/O

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Resource Lifetimes and Cleanup

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Resource Cleanup Pattern

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Multiple Resources

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Error Handling with Resources

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 18: Concurrency and Threading

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Threads and the Operating System

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Why OS Threads?

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Creating and Managing Threads

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Creating a Thread

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Thread Procedure Signature

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Joining Threads

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Destroying Threads

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Thread Pools and Task Queues

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Using `core:thread Pool`

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Thread Pool Advantages

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Task Data

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Mutexes, Read-Write Locks and Semaphores

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Mutex

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Read-Write Mutex

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Semaphore

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Wait_Group

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Condition Variables

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Atomic Operations and Lock-Free Programming

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Basic Atomics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Atomic Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Memory Ordering

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Compare-and-Swap

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

When to Use Atomics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Context in a Multi-Threaded World

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Thread Context Isolation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Sharing Data Between Threads

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Thread-Local Storage

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 19: Runtime Type Information and Reflection

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Typeid: Runtime Type Identifiers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Getting Typeids

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Comparing Typeids

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Typeids Are Opaque

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

The Reflect Package: Inspecting Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Type_Kind

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Inspecting Structs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Field Tags

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Procedure Reflection

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Field Tags and Metadata

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

JSON Tags

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Custom Tags

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Tag Parsing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Any Type and Type Erasure

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

The any Type

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Using any

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Type Assertion

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Use Cases for any

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Use Cases: Serialization, Debugging and Tooling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Serialization

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Generic Printing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Debugging and Logging

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Tooling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The Cost of RTTI

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Memory Overhead

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Runtime Overhead

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Performance Impact

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

When to Disable RTTI

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 20: Inline Assembly and Intrinsics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Intrinsics: Compiler-Hinted Operations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

base:intrinsics Package

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Atomic Intrinsics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

SIMD Intrinsics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

SIMD Operations in core:simd

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

SIMD Vector Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

SIMD Operations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

SIMD and Performance

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

The New Inline Asm Template System

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Basic Syntax

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Typed Operands

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Intel Operand Order

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Register Binding

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Cross-Platform Assembly

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

When Inline Assembly Is Useful

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Writing Portable Assembly

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Use Intrinsics When Available

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Abstract Assembly Behind Procedures

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Document Assembly Assumptions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Test on Target Hardware

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

When to Use Assembly vs High-Level Code

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Use High-Level Odin When:

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Use Intrinsics When:

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Use Inline Assembly When:

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Profile First

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Profiling Assembly Output

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Viewing Generated Assembly

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Using Compiler Explorer

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

What to Look For

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Chapter 21: Project 1 - A

Command-Line Binary File Parser

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Requirements and Design Decisions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

The File Format

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Requirements

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Design Decisions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Defining the File Format with Odin Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Header Type

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Entry Type

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Parser Result

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Parsing Logic and Error Handling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Opening and Reading the File

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Parsing the Header

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Parsing Entries

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Formatting Output

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Command-Line Argument Processing with core:flags

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Argument Definition

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Testing with Synthetic Input Files

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Test File Generator

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Error Case Tests

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Packaging and Distribution

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Project Structure

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Building

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Cross-Compilation (host toolchain permitting)

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Running

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 22: Project 2 - A Custom Memory Allocator

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Allocator Requirements and Interface

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The Allocator Interface

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Implementing an Arena Allocator

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Arena Structure

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Initialization

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Allocation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The Arena's Allocator Procedure

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Usage

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Implementing a Pool Allocator for Fixed-Size Objects

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Pool Structure

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Initialization

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

The O(1) Helpers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

The Pool's Allocator Procedure

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Integration with the Context System

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Scoped Allocator Override

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Using the Pool for a Specific Type

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Performance Measurement and Comparison

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Benchmark Harness

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Pool vs Default for Fixed-Size Objects

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Debugging and Validation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Validation Checks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Canaries

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 23: Project 3 - A

Data-Oriented Physics Simulation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Simulation Requirements and Data Model

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Requirements

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

The Physics

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Data Model

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Implementing SoA Particle Storage

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Initialization

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Using Odin's #soa Modifier

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

SIMD-Accelerated Force Calculations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Gravity and Drag with SIMD

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Integration with SIMD

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Fixed Time-Step Integration Loop

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

The Simulation Loop

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Main Entry Point

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Performance Analysis and Optimization

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Baseline: Scalar Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

SIMD Improvement

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Cache Efficiency

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Memory Bandwidth

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Optimization Ideas

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Visualization Options

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Text Output

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

OpenGL/Vulkan Rendering

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Export to File

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Chapter 24: Project 4 - A

Multithreaded Job System

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Design Requirements and Threat Model

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Requirements

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Threat Model

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Thread Pool Implementation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Core Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Initialization

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Worker Thread Procedure

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Submitting Jobs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Waiting for Completion

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Destroying the Pool

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Work Queue and Scheduling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguidetolow-levelprogramming>.

Fair Scheduling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguidetolow-levelprogramming>.

Prioritized Scheduling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguidetolow-levelprogramming>.

Thread-Local Storage and Context Management

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguidetolow-levelprogramming>.

Thread-Local Buffers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguidetolow-levelprogramming>.

Using Local Storage in Jobs

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguidetolow-levelprogramming>.

Synchronization and Correctness Guarantees

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Mutex Usage Rules

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Atomic Operations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Avoiding Deadlocks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Memory Safety

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Testing for Races and Deadlocks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Stress Testing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Deadlock Detection

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Using Thread Sanitizers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 25: Project 5 - A TCP Networking Service

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Service Requirements and Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Requirements

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Architecture

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Socket Programming with core:net

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Creating a Listening Socket

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Accepting Connections

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Connection Handling and Event Loop

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Connection Handler

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Request Processing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Building Responses

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Handling Multiple Concurrent Connections

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

One Thread Per Connection

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Using a Thread Pool

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Protocol Design and Message Framing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Line-Based Protocol

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Binary Protocol

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Resource Management and Graceful Shutdown

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Cleaning Up on Shutdown

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Running the Server and Graceful Shutdown

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Security Considerations

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Input Validation

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Connection Limits

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Buffer Sizes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

TLS/SSL

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Performance and Scaling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Non-blocking I/O

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Connection Keep-Alive

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Chapter 26: Build Systems, Optimization and Tooling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

The Odin Build System

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Project Layout

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Build Commands

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Build Modes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Incremental Builds

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Conditional Builds with #config

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Performance Optimization

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Compiler Optimization Levels

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Key Optimization Techniques

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

1. Cache-Friendly Data Layouts

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

2. Avoid Dynamic Allocation in Hot Paths

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

3. Use SIMD

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

4. Use Fixed Arrays Instead of Slices When Possible

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

5. Inline Small Procedures

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Optimizing Based on Profiles

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Profiling Tools

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

CPU Profiling with Spall

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Viewing Profiles

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Memory Profiling

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Time Measurements

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Using External Tools

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Debugging

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Debug Builds

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Using Debuggers

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

GDB

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

LLDB

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Visual Studio Code

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Assertions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Logging

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Memory Debugging

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Production Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Static Linking

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Minimal Executables

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Packaging

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

CI/CD

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Chapter 27: Testing, Debugging and Quality Assurance

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Odin's Built-in Testing Framework

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Writing Tests

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Testing Assertions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Test Structure

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Testing with Mocks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Property-Based Testing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Implementing Property Tests

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Testing Round-Trip Conversions

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Testing Patterns and Best Practices

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Arrange-Act-Assert

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Test Tables

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Testing Error Cases

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Performance Tests

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Integration Tests

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Advanced Debugging Techniques

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Debugging Concurrency Issues

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Reproducing Races

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Lock Ordering

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Debugging Memory Issues

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Use-after-free

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Memory Leaks

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Core Dump Analysis

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Static Analysis and Code Quality

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Compiler Warnings

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Code Formatting

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Linting

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Code Review Practices

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Continuous Quality

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Automated Testing in CI

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Test Coverage

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Regression Testing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Dependency Management

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Summary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Conclusion

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

What Makes Odin Unique

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Skills You Gained

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Next Steps

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

The Future of Odin

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Final Thoughts

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Appendix A: Language Quick Reference

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Literals

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Types

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Procedures

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Control Flow

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Operators

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Memory

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Meta Operators

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Attributes

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practicalguide to low-level programming>.

Appendix B: Compiler and Tool Reference

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

odin build

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Key Flags

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

odinfmt

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Odin Language Server (OLS)

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odinapracticalguidetolow-levelprogramming>.

Testing

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Useful Commands

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

Glossary

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.

References

This content is not available in the sample book. The book can be purchased on Leanpub at <https://leanpub.com/odina practical guide to low-level programming>.