Table of Contents

Foreword ix

Preface xi

1. Introduction 1
What Are Threads? 3
Concurrency Versus Parallelism 5
Single-Threaded JavaScript 6
Hidden Threads 9
Threads in C: Get Rich with Happycoin 10
With Only the Main Thread 11
With Four Worker Threads 13

2. Browsers 19
Dedicated Workers 20
Dedicated Worker Hello World 20
Advanced Dedicated Worker Usage 23
Shared Workers 25
Shared Worker Hello World 27
Advanced Shared Worker Usage 32
Service Workers 33
Service Worker Hello World 35
Advanced Service Worker Concepts 40
Message Passing Abstractions 43
The RPC Pattern 43
The Command Dispatcher Pattern 45
Putting It All Together 47

3. Node.js. 53
Before We Had Threads 54
The worker_threads Module 56
workerData 57
MessagePort 58
Happycoin: Revisited 60
With Only the Main Thread 60
With Four Worker Threads 63
Worker Pools with Piscina 65
A Pool Full of Happycoins 69

4. Shared Memory. 73
Intro to Shared Memory 74
Shared Memory in the Browser 74
Shared Memory in Node.js 77
SharedArrayBuffer and TypedArrays 79
Atomic Methods for Data Manipulation 84
Atomics.add() 85
Atomics.and() 85
Atomics.compareExchange() 86
Atomics.exchange() 86
Atomics.isLockFree() 86
Atomics.load() 86
Atomics.or() 87
Atomics.store() 87
Atomics.sub() 87
Atomics.xor() 87
Atomicity Concerns 88
Data Serialization 91
Booleans 91
Strings 93
Objects 94

5. Advanced Shared Memory. 97
Atomic Methods for Coordination 97
Atomics.wait() 98
Atomics.notify() 99
Atomics.waitAsync() 100
Timing and Nondeterminism 100
Example of Nondeterminism 100
Detecting Thread Preparedness 104
Example Application: Conway's Game of Life 106

Single-Threaded Game of Life 107
Multithreaded Game of Life 112
Atomics and Events 118

6. Multithreaded Patterns. 121
Thread Pool 121
Pool Size 122
Dispatch Strategies 124
Example Implementation 125
Mutex: A Basic Lock 131
Streaming Data with Ring Buffers 137
Actor Model 144
Pattern Nuances 145
Relating to JavaScript 146
Example Implementation 146

7. WebAssembly. 155
Your First WebAssembly 155
Atomic Operations in WebAssembly 157
Compiling C Programs to WebAssembly with Emscripten 159
Other WebAssembly Compilers 160
AssemblyScript 161
Happycoin in AssemblyScript 163

8. Analysis. 169
When Not to Use 169
Low Memory Constraints 170
Low Core Count 172
Containers Versus Threads 175
When to Use 176
Summary of Caveats 181

Appendix. Structured Clone Algorithm. 183

Index. 187