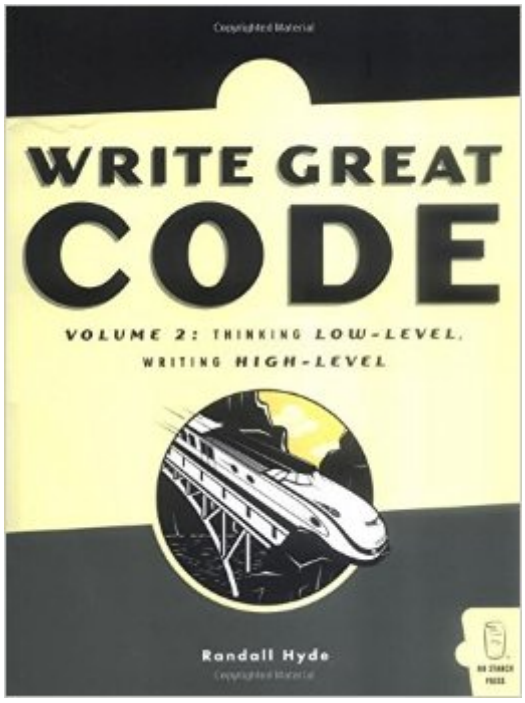
Write Great Code, Volume 2: Thinking Low-Level, Writing High-Level



Synopsis

The second volume in the Write Great Code series supplies the critical information that today's computer science students don't often get from college and university courses: How to carefully choose their high-level language statements to produce efficient code. Write Great Code, Volume 2: Thinking Low-Level, Writing High-Level, teaches software engineers how compilers translate high-level language statements and data structures into machine code. Armed with this knowledge, a software engineer can make an informed choice concerning the use of those high-level structures to help the compiler produce far better machine code--all without having to give up the productivity and portability benefits of using a high-level language.

Book Information

Series: Write Great Code (Book 2)

Paperback: 640 pages

Publisher: No Starch Press; 1 edition (March 18, 2006)

Language: English

ISBN-10: 1593270658

ISBN-13: 978-1593270650

Product Dimensions: 7 x 1.5 x 9.2 inches

Shipping Weight: 2.3 pounds (View shipping rates and policies)

Average Customer Review: 4.2 out of 5 starsÂ Â See all reviewsÂ (10 customer reviews)

Best Sellers Rank: #881,535 in Books (See Top 100 in Books) #57 inÂ Books > Computers & Technology > Programming > Languages & Tools > Assembly Language Programming #130 inÂ Books > Computers & Technology > Computer Science > AI & Machine Learning > Machine Theory #301 inÂ Books > Textbooks > Computer Science > Object-Oriented Software Design

Customer Reviews

I earned my Computer Science degree several years ago. During my studies I learned languages like Assembly, Modula-2, C and C++ and even a little VB in a computer graphics class. I learned quite a bit about how computers work at the binary level and even got into some Electrical Engineering topics.As you might expect, I don't use any of those languages in my career today. It's all Java, LotusScript, and other high-level languages for me these days. I rarely find myself shifting bits or thinking about logic gates. Does this mean my CS degree was a waste of time? Absolutely not. Although it would have been nicer if I had been learning Java... :-)It's not the languages I learned that gave the education its value. It's the algorithms, patterns and logic that have come to

my aid again and again. An understanding of what's going on inside the computer can be very helpful at times. That's why I was so interested in reading Randall Hyde's book: Write Great Code, Volume 2: Thinking Low-Level, Writing High-Level.This book teaches the following concepts (Chapter 1 excerpt):- Why it's important to consider the low-level execution of your high-level programs- How compilers generate machine code from high-level language (HLL) statements- How compilers represent various data types using low-level, primitive, data types- How to write your HLL code to help the compiler produce better machine code- How to take advantage of a compiler's optimization facilities- How to "think" in assembly language (low-level terms) while writing HLL codeThe general goal of this book is to teach you how to think like a compiler so you can anticipate what the compiler will do with your code. Randall Hyde is also the author of The Art of Assembly Language.

Back in "the day", you really couldn't write high-level code without at least some exposure at some point to lower-level code, like Assembler. Now, you can pretty much be completely ignorant of what happens in your Java or VB code "under the covers". But that doesn't mean you can't benefit from understanding how your compiler turns your readable code into machine-readable operations. Randall Hyde does an excellent job in explaining all this in his book Write Great Code Volume 2: Thinking Low-Level, Writing High-Level.Contents: Thinking Low-Level, Writing High-Level; Shouldn't You Learn Assembly Language?; 80x86 Assembly for the HLL Programmer; PowerPC Assembly for the HLL Programmer; Compiler Operations and Code Generation; Tools for Analyzing Compiler Output; Constants and High-Level Languages; Variables in a High-Level Language; Array Data Types; String Data Types; Pointer Data Types; Record, Union, and Class Data Types; Arithmetic and Logical Expressions; Control Structures and Programmatic Decisions; Iterative Control Structures; Functions and Procedures; Engineering Software; A Brief Comparison of the 80x86 and PowerPC CPU Families; Online Appendices; IndexThis is the type of book that will really excite you if you're wondering why a nested if statement performs differently than a case statement. Hyde explains basic compiler theory, and applies that to how your compiler of choice decides on optimization strategies. It's impractical to get a program optimized for all factors, like code size and speed, but there are reasonable trade-offs as well as compiler options you can use to prioritize one factor over another. You also don't have to be completely conversant with Assembler in order to work through this book.

I read the first volume of this book, and it was a great, informative read. After volume 2, I have this

to say:This book is not a cookbook for writing better code. Hyde explains why certain programming constructs are better than others (and in what cases), and backs it up with evidence from the assembly code (that is the entire premise of the book). Finally, solid proof of (and against) what I've been hearing all along from instructors and other programmers on message boards or face-to-face communication. That in mind, it would have been nice to have a summary of the tips at the end of the chapters, or the end of the book, as a quick-reference kind of thing. These concepts are the perfect thing to consider when fine-tuning your code. I take the stance that if you fine-tune as you go, you have less work later, so I took notes as I read and have started implementing changes for the better (with evidence that it is better) in some of my coding.Does anyone write code in Pascal anymore? Seriously? Hyde discusses examples in C/C++ and Pascal for the most part (favoring Pascal, by my estimation), so it is nice that the book is language-independent for the reader. The assembly examples in the book are in 80x86 and PowerPC...I think it would have been better to release two versions of this book, one where the assembly is 80x86 (because it is so ubiquitous, if for no other reason), and then another where the assembly is in PowerPC assembly. I didn't pick this up to become an assembly language programmer, and quite frankly, the PowerPC examples just confused me. It looks like my cat stepped on my keyboard and it appeared in print.

protein, low carbohydrate foods, low carb, low carb cookbook, ... Ketogenic Diet to Overcome Belly Fat) Low Carb Casseroles BOX SET 2 IN 1: 38 Super Satisfying Low Carb Casseroles That Will Amaze Your Family: (low carbohydrate, high protein, low carbohydrate ... diet for dummies, low carb high fat diet) Low Carb Diet BOX SET 3 IN 1: 30 Low Carb Snacks + 21 Low Carb Casseroles + 20 Low Carb Soups: (low carbohydrate, high protein, low carbohydrate foods, ... Ketogenic Diet to Overcome Belly Fat) Low Carb Diet: 15 Healthy And Delicious Low Carb Salads To Lose Weight Fast: (low carbohydrate, high protein, low carbohydrate foods, low carb, low carb ... Ketogenic Diet to Overcome Belly Fat) Low Carb Mexican Recipes: 25 Of Your Favorite Mexican Recipes Made Low Carb!: (low carbohydrate, high protein, low carbohydrate foods, low carb, low ... Ketogenic Diet to Overcome Belly Fat) Low Carb Diet - Top 200 Low Carb Recipes Cookbook: (Low Carb, Budget Cookbook, Low Carb Diet, Low Carb Recipes, Atkins Diet, Low Carb Slow Cooker Recipes, Low Carb Living) Writing : Novel Writing Mastery, Proven And Simple Techniques To Outline-, Structure- And Write A Successful Novel ! - novel writing, writing fiction, writing skills - Low Carb: Low Carb Diet for Beginners - How to Lose 7 Pounds in 7 Days with Low Carb & High Protein Diet Without Starving! (low carbohydrate, high protein, ... carb cookbook, ketogenic diet, paleo diet) Low Carb Diet. Slow Cooker Recipes: 25 Delicious Low Carb Dinners To Lose Weight Fast: (low carbohydrate, high protein, low carbohydrate foods, low carb, ... Ketogenic Diet to Overcome Belly Fat) Low Carb Cookies: 23 Best Low Carb Cookie Recipes To Maintain Your Healthy Eating Habits: (low carbohydrate, high protein, low carbohydrate foods, low ... Ketogenic Diet to Overcome Belly Fat) Low Carb: Low Carb, High Fat Diet. The Winning Formula To Lose Weight (Healthy Cooking, Low Carb Diet, Low Carb Recipes, Low Carb Cookbook, Eat Fat, Ketogenic Diet) Positive Thinking: How to Eliminate Negative Thinking and Gain Success, Health and Happiness Through Positive Thinking and Self-empowering Affirmations (Positive Thinking Everyday Book 1)