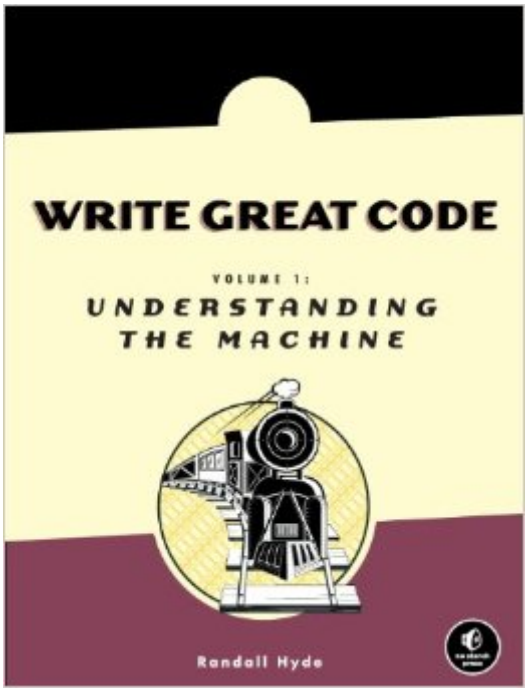
Write Great Code: Volume 1: Understanding The Machine



Synopsis

If you've asked someone the secret to writing efficient, well-written software, the answer that you've probably gotten is "learn assembly language programming." By learning assembly language programming, you learn how the machine really operates and that knowledge will help you write better high-level language code. A dirty little secret assembly language programmers rarely admit to, however, is that what you really need to learn is machine organization, not assembly language programming. Write Great Code Vol I, the first in a series from assembly language expert Randall Hyde, dives right into machine organization without the extra overhead of learning assembly language programming at the same time. And since Write Great Code Vol I concentrates on the machine organization, not assembly language, the reader will learn in greater depth those subjects that are language-independent and of concern to a high level language programmer. Write Great Code Vol I will help programmers make wiser choices with respect to programming statements and data types when writing software, no matter which language they use.

Book Information

Paperback: 440 pages

Publisher: No Starch Press; 1st edition (October 25, 2004)

Language: English

ISBN-10: 1593270038

ISBN-13: 978-1593270032

Product Dimensions: 7 x 1.1 x 9.2 inches

Shipping Weight: 1.9 pounds (View shipping rates and policies)

Average Customer Review: 4.5 out of 5 starsÂ Â See all reviewsÂ (23 customer reviews)

Best Sellers Rank: #696,175 in Books (See Top 100 in Books) #46 inÂ Books > Computers & Technology > Programming > Languages & Tools > Assembly Language Programming #122 inÂ Books > Computers & Technology > Computer Science > AI & Machine Learning > Machine Theory #1913 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Development

Customer Reviews

As computers have gotten smaller and faster, developers have become more and more removed from the lowest levels of programming. Randall Hyde's new book Write Great Code - Volume 1: Understanding The Machine (No Starch Press) will help you get back to the basic levels of how computers work and how that affects your programming.Chapter List: What You Need To Know To

Write Great Code; Numeric Representation; Binary Arithmetic And Bit Operation; Floating-Point Representation; Character Representation; Memory Organization And Access; Composite Data Types And Memory Objects; Boolean Logic And Digital Design; CPU Architecture; Instruction Set Architecture; Memory Architecture And Organization; Input And Output (I/O); Thinking Low-Level, Writing High-Level; ASCII Character Set; IndexIt used to be you couldn't program at all without knowing this material. The design of a program was tied closely to the machine architecture, and it drove the instruction set and the overall programming decisions. But now the higher-level programs have made it easier for mere mortals to write a program and be completely oblivious to how a CPU executes an instruction or loads data from memory. Hyde goes into great detail on all the instructional design and theory, and I'd venture to guess that a very small number of programmers (and I'm not one of them) know most of this information. The assumption is that you'll know at least one procedural language (like C, C++, BASIC, or assembly). He rotates examples among C, C++, Pascal, BASIC, and assembly so as to keep the examples as language-neutral as possible. The goal when you finish the reading is that you should understand exactly how the architecture of a CPU affects your program, and how to make programming decisions that will lead to efficient programs. This volume will be followed up by another book titled Think Low-Level, Write High-Level. For me, I think this is where a lot of this information will come together.Foundational information presented in great detail, and a book that all serious developers should take the time to read and understand.

This is a great book but I have to disagree with the overall viewpoint. I've been doing embedded programming for a while and if that's all I'd ever done I would totally agree that understanding low level concepts helps write better code. However, I also write a lot of code in C#. People who normally use high level languages such as C#, VB.Net, or JAVA are probably not going to benefit much from this book. These languages are so far abstracted from the hardware level that the concepts are hard to apply anywhere. On the other hand, if you still use malloc on a daily basis, you need to read the book :) Anyway, the book is easy to read and I never caught any errors. If you want to learn about computers at a low level, then this is a great book to start with!

As new computer languages arise that have more power, like Java and C#, have you noticed something? Often, someone might learn programming without ever having to know about the architecture of a von Neumann machine. Yet most computers since World War 2 have this design at their very core.Hyde fills in this gap in the education. At one level, you should read it for "culture". It

explains the basis of programming. Granted, for most of us, there is often no direct need for understanding how binary arithmetic is implemented. Or why registers can speed up performance. And what is cache memory, really? We finesse our ignorance by invoking libraries that subsume these details.The material that Hyde explains may occasionally be of use. What if you need to write some of these low level procedures in assembler, to reduce a bottleneck? After using a profiler on your runtime code to find the key routines, do you have any idea how to improve matters?Even out of pure intellectual curiosity, you should know what happens at the silicon. Or are you just a wage slave? Programming because you have to? A good programmer who loves to program should know this material. Also, out of pure self interest, you should always burnish your programming skills. To separate you from your peers.

A great programmer has both a compiler and a CPU in his head. You have to understand how a machine operates to understand how you can get yourself out of trouble if you have a problem that you don't understand. Particularly when you are using a systems language like C, C++ or assembler.This book provides an in-depth understanding of the working of a CPU. And it does it in a well written and organized way with very effective use of illustrations. This is not the assembler book you remember. This book is targeted at systems level programmers who need to understand the machine in order to make the best use of it.Given that many programmers start by learning Java, and learn C as their low level language, I can see there being a good market for this book. If you are working on a large C or C++ application, or are writing C libraries for Java, and you don't understand the basics (memory management, stacks, bit shifting, assembler opcodes, etc.) you should get this book.

Write-and-Learn Sight Word Practice Pages: Engaging Reproducible Activity Pages That Help Kids Recognize, Write, and Really LEARN the Top 100 High-Frequency Words That are Key to Reading Success How to Write the Perfect Personal Statement: Write powerful essays for law, business, medical, or graduate school application (Peterson's Perfect Personal Statements) You Can Write a Column (You Can Write It!) How to Write Better RÃƒÂ©sumÃƒÂ©s and Cover Letters (How to Write Better Resumes and Cover Letters) Resume: How To Write A Resume Which Will Get You Hired In 2016 (Resume, Resume Writing, CV, Resume Samples, Resume Templates, How to Write a CV, CV Writing, Resume Writing Tips, Resume Secrets) This book will teach you how to write better: Learn how to get what you want, increase your conversion rates, and make it easier to write anything (using formulas and mind-hacks) Songwriting 101 (2nd Edition): Inspiration, Tips, Tricks, and Lessons for the Beginner, Intermediate, and Advanced Songwriter (lyrics, writing songs, songwriter, ... write music, write lyrics, song writing) How to Write Songs on Keyboards - A Complete Course to Help You Write Better Songs Book/online audio (Softcover) How to Write It, Third Edition: A Complete Guide to Everything You'll Ever Write Launch to Market: Easy Marketing For Authors (Write Faster, Write Smarter Book 4) Guys Write for Guys Read: Boys' Favorite Authors Write About Being Boys The Bread Lover's Bread Machine Cookbook: A Master Baker's 300 Favorite Recipes for Perfect-Every-Time Bread-From Every Kind of Machine More Bread Machine Magic : More Than 140 New Recipes From the Authors of Bread Machine Magic for Use in All Types of Sizes of Bread Machines