The Elements Of Computing Systems: Building A Modern Computer From First Principles
Synopsis

In the early days of computer science, the interactions of hardware, software, compilers, and operating system were simple enough to allow students to see an overall picture of how computers worked. With the increasing complexity of computer technology and the resulting specialization of knowledge, such clarity is often lost. Unlike other texts that cover only one aspect of the field, The Elements of Computing Systems gives students an integrated and rigorous picture of applied computer science, as its comes to play in the construction of a simple yet powerful computer system. Indeed, the best way to understand how computers work is to build one from scratch, and this textbook leads students through twelve chapters and projects that gradually build a basic hardware platform and a modern software hierarchy from the ground up. In the process, the students gain hands-on knowledge of hardware architecture, operating systems, programming languages, compilers, data structures, algorithms, and software engineering. Using this constructive approach, the book exposes a significant body of computer science knowledge and demonstrates how theoretical and applied techniques taught in other courses fit into the overall picture. Designed to support one- or two-semester courses, the book is based on an abstraction-implementation paradigm; each chapter presents a key hardware or software abstraction, a proposed implementation that makes it concrete, and an actual project. The emerging computer system can be built by following the chapters, although this is only one option, since the projects are self-contained and can be done or skipped in any order. All the computer science knowledge necessary for completing the projects is embedded in the book, the only pre-requisite being a programming experience. The book’s web site provides all tools and materials necessary to build all the hardware and software systems described in the text, including two hundred test programs for the twelve projects. The projects and systems can be modified to meet various teaching needs, and all the supplied software is open-source.

Book Information

File Size: 6531 KB
Print Length: 344 pages
Publisher: The MIT Press (January 25, 2008)
Publication Date: January 25, 2008
Sold by: Digital Services LLC
Language: English
ASIN: B004HHORGA
I highly recommend this book if you are interested in learning about computer science. The book is organized around the idea of building a computer from the fundamental logic gates up--starting with the hardware (combinational logic gates, arithmetic logic units, sequential logic gates, the CPU and memory) and then through the software hierarchy (starting with the machine language, and working up through the assembler, a virtual machine, a compiler for a high-level language, and an operating system). As a "by-product," one learns, by very relevant examples, many fundamental concepts of computer science. You can just read the book, but the best idea is to follow the authors’ advice and do the projects where you implement every necessary piece of the computer system yourself. The projects are all very well organized. All the software necessary to emulate any part of the computer is available for free download from the authors’ web-site. It all works beautifully. If you want to skip any of the projects, you can, because the software is organized in such a way that it will use built-in modules instead of the ones you built if necessary. The authors seem to have extensively tested the whole approach through the courses they have taught using this material. I also noticed that Harvard's Computer Science 101 course is being taught based on this book. I have been using the book for self-study with absolutely no problems--in fact I have never had such a great experience with a self-study course. All you need is a Windows or Linux (edit: Mac OS X works fine too) computer and access to the internet, and you can give yourself a wonderful education in computer science. In terms of prerequisites, you only really need to have some experience with programming (e.g.

When I say "survey," I mean a wide-ranging view of what goes into a computing system. This unique book goes into more depth than the word suggests, though, since it presents every level as
a project for the student to carry out. This starts with the processor: the authors present an elegant but very stripped-down instruction set, slightly reminiscent of the PDP-8, and an equally stripped-down hardware description language (HDL). The exercise is to implement that processor using that HDL, and verify it using a simulator the authors provide. Next, the student implements the assembler for that instruction set, an interpreter in the spirit of the Java Virtual Machine, a compiler, and a simple operating system. Although each project could be a term course in itself, the authors display a real knack for extracting the essentials of each and boiling them down to a minimal but functional kernel. The results, although they might be toy systems, demonstrate the framework around which larger, industrial systems would be built.

I've taught HDL-based logic design, operating systems, and object-oriented design. Each level of system implementation makes sense only in terms of the levels above and below it, but each is normally taught in isolation. This leaves an odd lack of context and motivation. It never explains to a processor designer what hardware support an OS needs, never explains to the OS implementor what the hardware can (and can't) do, or what the application developer requires. This philosophy even lets computer science students graduate in ignorance of or disdainful of the hardware on which their whole career depends. And, at every level, crucial basics like "what is a stack frame?" go unexplained and unexplored.

Download to continue reading...