Principles Of Concurrent And Distributed Programming (2nd Edition)
**Synopsis**

The latest edition of a classic text on concurrency and distributed programming; from a winner of the ACM/SIGCSE Award for Outstanding Contribution to Computer Science Education.

**Book Information**

Paperback: 384 pages  
Publisher: Pearson; 2 edition (March 6, 2006)  
Language: English  
ISBN-10: 032131283X  
Product Dimensions:  6.8 x 0.9 x 9.1 inches  
Shipping Weight: 1.4 pounds (View shipping rates and policies)  
Average Customer Review:  3.8 out of 5 stars  
Best Sellers Rank: #310,604 in Books (See Top 100 in Books)  
#34 in Computers & Technology > Programming > Parallel Programming  
#2264 in Computers & Technology > Programming > Languages & Tools  
#4905 in Textbooks > Computer Science

**Customer Reviews**

This book is a poor choice for an introductory concurrency textbook. The concepts of concurrency and the mechanisms used to implement it are not that difficult to understand, however, the author’s overly terse and poorly presented core material renders an otherwise simple subject incomprehensible. Some chapter introductions read reasonably well; however the author routinely dismisses the completion of various proofs as trivial, rendering self-study virtually impossible if something is not clear. If the book is used in a course with a poor instructor, you’ll waste a lot of time trying to learn from the book, and will need to find additional resources. A primary drawback is the author’s heavy reliance on mathematical proofs (admittedly a core part of proving concurrency works) is unsupported by an adequate quantity fully completed non-trivial proofs properly organized and documented to be suitable for teaching the process and logic the author used to reach a given conclusion. As with many mathematical texts, the author assumes that the audience, understands all of the steps he has skipped to get from A to B. The style is suitable for technical papers but not texts intended to teach the more abstract subject matter. A second issue is the author’s contrivance of a pseudo-code shorthand that is used throughout the book. While the pseudo code is reminiscent of many mathematical notations as well as the the pseudo code notation used in Knuth’s ‘The Art of Computer Programming’, the teaching aspects would be better served by providing inline
comments, and other cues that are generally expected in modern software. Knuth’s explanations are a benchmark in educational texts.

Overall I think this book is good, but contains a ton of unnecessary imperfections that could have easily have been avoided. If you are able to ignore those parts and not let them ruin your mood, I think you will enjoy this book and learn efficiently from it. I want to add that my grade is closer to 2 stars than to 4 stars. With some improvements, I think the book could easily deserve 4 or even 5 stars.

**MERITS:** It has a nice mathematical/rigorous view of the subject and emphasizes the important things, which are the principles and concepts of concurrent programming. I am greatly in favor of this philosophy, because the first thing you want to do when learning a new subject is to get the panoramic view of what are the problems and techniques of the field, and how do all the concepts relate to each other. There is also merit in isolating what is actually unique about concurrent programming, and avoid bloated 1000-page works that are filled with things you already know about other subjects, e.g. general programming techniques. Also focusing on pseudocode instead of cluttering the algorithms with syntax of particular languages is a benefit. Ben-Ari succeeds in writing a pretty friendly page-turner which will have you understand what concurrent programming is about, and most of the central constructs, in a matter of a couple of weeks.

**DEMERITS:** At the level of details, the book is, unfortunately, not well thought out. There are tons of sentences that are vague, unnecessarily complicated or ambiguous. In most of these spots it is possible to rule out all incorrect interpretations, or to fill in the gaps, given enough time.

Download to continue reading...

Dmca