

## BOOKS

*Books should be sent to the editor-in-chief. Selected books, which are within the scope of SCP and are not proceedings, will be reviewed. Others may be mentioned.*

*Principles of Concurrent Programming.* By M. Ben Ari. Prentice-Hall International, London, 1982, 172 pp.

There are two good textbooks on the subject of operating systems which devote a lot of attention to concurrent programming: Brinch Hansen's *Operating Systems Principles* and Holt's et al: *Structuring Concurrent Programming with Operating Systems Applications*. The author of the book under review rightly decided that the time is ripe to offer the readers a textbook entirely devoted to the subject of concurrent programming. The book is appropriate for a semester course on the subject. As the author observes no prerequisites other than computer science maturity are required.

The book consists of seven chapters and an appendix. After two introductory chapters focusing on the nature and origins of concurrent programming the author discusses the problem of mutual exclusion and presents Dekker's solution. The presentation is based on the original Dijkstra's article while more recent papers (like Lamport's bakery algorithm) are relegated to the exercises. This creates an erroneous impression that not much has happened in this area since 1968. Various issues like solutions for  $n$  processes, distributed solutions, concurrent reading and writing and solutions for the case when processes can fail are introduced as secondary problems or even not mentioned at all. The only innovation of this chapter is that the correctness proof of Dekker's solution is presented in a rigorous yet informal way using temporal logic.

Next chapter is devoted to the study of semaphores. The presentation is once again based on Dijkstra's original article but this time more references to the recent literature can be found in the text and exercises. In particular simulation of a general semaphore by a split binary semaphore and the cigarette smoker's problem are discussed in the exercises.

Next chapter is about monitors. The presentation is this time based on the original article of Hoare. Unfortunately exercises are more about semaphores than monitors. In particular none of the issues studied after the introduction of monitors—like conditional **wait** or nested monitor calls are even mentioned. Topics discussed so far can be found in one way or the other in the books mentioned before: those of Brinch Hansen and Holt et al. The first novel chapter in this regard is the one about

the ADA rendezvous concept. The presentation is on the other hand very brief and sketchy.

The final chapter is devoted to the presentation of the problem of dining philosophers. Various attempts of solutions and correct solutions are coded using the semaphores, monitors and conditional critical regions. The book concludes with a long appendix in which a listing of a simple system allowing a concurrent execution of simple Pascal programs augmented by semaphores is presented. The system itself is written by N. Wirth.

A novel aspect of the book is the presentation of the proofs of the programs using a semi-formal reasoning based on temporal logic. The proofs are convincing and very well presented. They take in total 9 pages and provide the best informal introduction to the subject of temporal logic one could think of.

Apart from these 9 pages and the short, 16 pages long chapter on the ADA Rendezvous the whole book could have been written some six–seven years earlier. Rare pointers to the literature on the subject written after 1975 do not change this overall impression.

It is a pity that there is no chapter about CSP. This would allow the author to focus more on some of the problems inherent to distributed processing like the problem of distributed termination of Francez only cited in the bibliography. Such a chapter would improve the balance between parts devoted to multiprogramming and distributed processing. Also an introduction of chapter on CSP would allow a critical comparison of two different tools introduced for the same purpose—CSP and the ADA tasking, an approach so successfully exploited in the book in the case of multiprogramming.

On the other hand it should be clearly stated that the book is very well if not exceptionally well written. It makes really a pleasant reading. Regular references to chivalrous esquimos trying to work out appropriate protocols to enter their igloos or to exchange meat for sandwiches help to visualize the addressed problems and to understand the proposed solutions. Writing about concurrent programming is not an easy task and the author succeeded in this domain superbly presenting the material in a perfectly clear and convincing manner. The only objection one could have is that some of the exercises of the form “Study the following program . . .” without any additional comments can only frustrate the reader rather than to enlighten him. Studying ‘raw’ concurrent programs can be really a very difficult task.

Summarizing, this is a very nicely written book which offers a bit too old and uneven view of the subject. Those wishing to get complementary information on the subject are recommended to consult Andrews and Schneider’s article “Concepts and notations for concurrent programming”, *Comput. Surveys*, **15** (1) (1983).

Krzysztof R. APT  
*LITP, Université Paris 7*  
*Paris, France*