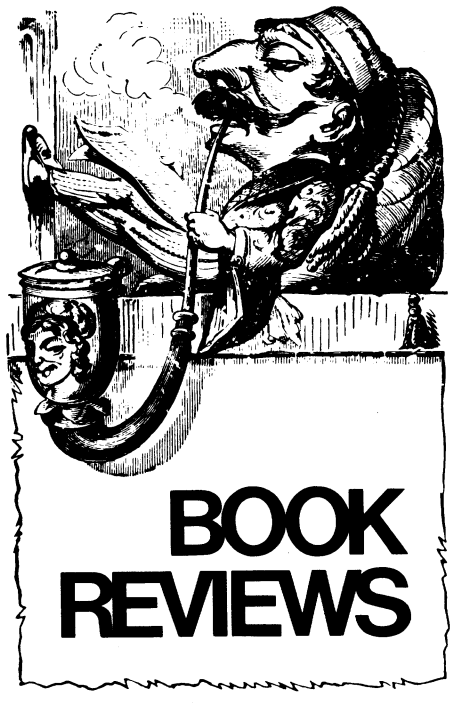
*Books submitted for review should be sent to COMPUTER Magazine, 5855 Naples Plaza, Suite 301, Long Beach, CA 90803, attention Book Reviews Editor.*



# BOOK REVIEWS

**B75-7** *Operating Systems* – D. C. Tsichritzis and P. A. Bernstein (New York: Academic Press, 1974, 298 pp., $13.50)

Of the recent spate of textbooks on operating systems, only one is written by a person who has actually built a working system.[1] This book is not that one, and it shows.

The authors have written a somewhat superficial survey of the field. They have followed closely the recommendations of the Cosine committee on teaching operating systems[2] (of which both the reviewer and one of the authors were members), and this has ensured reasonably complete coverage. Unfortunately, it has not protected them against a large number of errors of detail.

There are two major parts, on principles and techniques. Under the former heading are topics for which some unifying theory exists: processes and processor allocation, memory management, and virtual memory. Under the latter fall more pragmatic topics: input/output and file systems, protection, design and implementation methodology, and descriptions of two example systems. There are also appendices on data structures and computational structures.

The treatment of processes and scheduling is adequate, but suffers by comparison with Brinch Hansen's authoritative handling of these topics.[1] Memory mapping techniques are described competently, except for a misleading treatment of hardware support for segmentation, and the fact that (like many others) the authors think dynamic linking and segmentation are inseparably connected. The discussion of memory hierarchies in Chapter 5 is the low point of the book: careless statements about hardware, a confusing survey of replacement rules, and a handwaving section on the performance of paging algorithms leave nothing to recommend. The complete avoidance of any quantitative analysis, either of scheduling or of memory hierarchies, carries a good thing a little too far; students should learn what a surprising amount of insight can be obtained from very crude models.

The surveys of file systems and protection are adequate, and the chapters on design and implementation contain a good deal of interesting material. The strength in this area reflects the fact that the authors shared in the design and preliminary implementation of the Sue system. The best part of the book is the descriptions of the Sue and Venus systems, from which the careful reader can learn much about how to build scattered parts into a coherent whole (especially if he follows up the references).

There is a good annotated bibliography, a cursory appendix on list structures, and another appendix with a concise treatment of Petri nets, computational schemata, and reusable resource graphs. Finally, there is a valuable description of how to set up a "toy operating system" project from which students can learn something about the realities of system-building.

In summary, this book is a pleasant survey, best where it deals with pragmatic matters, but neither authoritative nor profound.

Butler W. Lampson
Xerox Research Center
3180 Porter Drive
Palo Alto, CA 94304

1. P. Brinch Hansen, *Operating System Principles.* Englewood Cliffs, N. J.: Prentice-Hall, 1973.

2. P. J. Denning et. al., *An Undergraduate Course on Operating Systems Principles.* Commission on Education, National Academy of Engineering, 2102 Constitution Ave., Washington, D. C.