Greek letters. The use of the same type of letters for functions of different level make simple concepts hard to grasp.

The reviewer at times finds it most frustrating trying to parse some of the sentences. Some of them have such a complex structure that the reviewer just had to give it up. Perhaps this is unavoidable since this book is a translation from a German text. There are several broken sentences and, of course, some typographical errors. But they are mostly harmless.

In conclusion, the reviewer is pleased to introduce this book as one to be used in a graduate seminar in automata theory and as a reference book for research purposes. The bibliography includes a rather exhaustive list of research works in automata theory published up until about 1969. This can be very useful for people interested in doing research in this area.

REFERENCE


ARTHUR T. POE
Temple University

* Also known as A. T. Pu.


This is the first book to cover the topic of interactive computer graphics. I found the book to be both very useful and very disappointing.


The authors suggest that an eight-week undergraduate course could be built around the introduction and first two parts; adding the third section could lead to a more substantial undergraduate program, and a full year’s graduate course could be built on the full book.

It is not terribly clear what the prerequisites should be. The first section on display devices is not written carefully. An unwary student would need help to stay out of the traps that the poor definitions and explanations present. My reading of the material describing how cathode ray tubes and storage tubes worked just served to make me alternately frustrated and angry, for example, the total cathode description follows.

1.3.1 CATHODE

The cathode is a small metal cylinder which is heated by an enclosed filament to a temperature at which it emits electrons.

1.3.2 CONTROL GRID

Do the authors wish the students to infer that the flow of electrons is a step function of temperature? From a metal surface?

The bibliographic listings and references served to heighten both reactions. I now realize full well this is primarily a book on programming with a computer scientist’s view of computer graphics.

It is not necessary that a programmer understand the physical operation of displays or display devices and this book does nothing to help gain that understanding. For that reason, I feel that the first section could well be deleted, greatly abridged, or rewritten to do justice to its material. It seems to me that any serious physical science or engineering student who would take this course should know more about the topic of CRT’s than this section offers. Others who would take the course deserve a better introduction, and a much better bibliography. Certainly the text by Kazan and Knoll on “Electronic Image Storage” should at least be offered to serious students as a source of more reliable information on storage tubes.

Fortunately, the book gets better as it begins to concern itself with programming-like material. Even so it tends to talk in terms that have never really been defined. Early on it offers examples in a low level language. A paper clip at the appropriate appendix will ease the reader’s trouble in trying to translate the language used.

In this early section much is made of the use of a push down stack, a stack pointer, the use of instruction “push,” “jump,” “pop-jump,” “push-jump.” These are used often enough that one must either figure it all out or put the book aside, but the book itself really doesn’t either define or explain in any one paragraph what each term is or what it means.

The above criticisms make me believe it would be difficult to teach the undergraduate courses without a great deal of supplementary material to bring everything to a focus.

Part Two: Display Files begins to be more informative and more readable.

Part Three: Interactive Graphics is well done and fun to read. I enjoyed the book from here on. For example, Chapter 9, “Graphic Input Devices,” is well written. The descriptions of the various input devices from the “SRI Mouse” to the Sylvania Tablet and various light pens were very readable.

I must admit my joy somewhat diminished at the beginning of Part Five—“Graphics Systems.” But there is, I’m sure, an audience that will find the treatment here more attuned to their style than to mine.

This book is clearly uneven in its treatment. For one who must learn the topic to be able to use interactive computer graphics in his work, the task to learn what is needed from this book does not appear easy.

The difficulty lies about equally in the material as presented in the earlier chapters and the lack of good up-to-date references in several areas. The bibliography, for example, offers a few papers on display quality which range from 1942–1952 and one from 1968 that is not pertinent. I believe the student deserves better.

The book is a first. If you are a skillful programmer, and want to learn interactive graphics, this book is for you.

LUCIEN M. BIBERMAN
Institute for Defense Analyses
Arlington, Va. 22202

(Maidenhead, England: Infotech Information, Ltd., 1972, 563 pp., individual volume—$95.00, subscriptions available for the 8 volumes of series 1 and of series 2 to be issued at $590.00 per series.)

This is another in a series of state-of-the-art reports from Infotech. It consists of edited discussions and presentations plus invited papers. Because a volume of this price will most likely not be sent out on approval, I think it important to review briefly the contents in detail before commenting on their value.

There are three main sections: analysis, presentations, and invited papers. There is also an annotated bibliography and indices, both to this volume and to others in the series.

The analysis section consists of eight chapters ranging in level from “The Theory of Operating Systems” to “Job Control Language.” In fact, these chapters are merely bits and pieces of transcribed discussions interspersed with editorial comments by the volume editor. They are intended to serve as introduction and pointers to the more detailed information contained in the rest of the volume.

The presentations section consists of the following papers.

“Operating Systems: What Should Today’s User Expect?” by
D. F. Hartley. Concentrates on functional specifications of operating systems.

"Front End Systems," by P. C. Poole. Describes various front-end architectures for handling interaction, job control, and job preparation.

"Building on to an Operating System," by J. I. Strauss. Brief discussion of Unix and how of adding function to an existing operating system.

"The Use of Simulation in the Design of a Multi-Computer Operating System," by G. K. Hutchinson. The particular system designed drives machine tools and interactive design terminals satellited to an IBM 1800.

"The Interaction Between Operating Systems and Machine Architecture," by D. Howarth. Explores some of the influences of these interactions on system structure and on the functions available to the user.

"Operating Systems for a Range of Computers," by B. J. Moore. Describes the operating system family designed for the ICL series of computers.


"Operating Systems for Real Time," by P. R. Cox. Brief overview of some of the characteristics of real-time systems.


"The Development of a Special Purpose Operating System," by D. W. Barron. Describes a system for an ICL 1900 that is special purpose in the sense that it was designed to exploit a particular machine with a particular profile of user jobs.

"Protection and Access Control in Operating Systems," by B. W. Lampson. Describes some of the problems involved with providing protection. This is one of the more technical presentations.

The invited papers consist of the following:

"Discussion of LP 70 as a Language for Systems Implementation," by R. Aslanian and R. Recio. Describes a particular language. Would have been much better had they discussed some of the general problems dealing with implementation languages.

"Deadlocks in Computer Systems," by E. G. Coffman, Jr. This is a cut down version of his paper in Computing Surveys 3, 2. It is a rather good survey.


"GEORGE 3, the Compleat Operating System," by G. B. Newell. Description of another system.


"Foundation Software for Real Time," by D. E. Rimmer. Foundation software shows characteristics of the underlying hardware rather than of the functions performed. Not particularly enlightening.

"Job Control Languages: What the User Really Requires," by H. J. Weegenaar. Discussion of JCL characteristics and a plea for more rationalized forms.

The bibliography, prepared by Peter Denning, is annotated and is useful.

Overall, this volume is a reasonably good survey of the current state of the art of operating systems at the design and architectural level. It does not contain much detailed technical information (e.g., scheduling algorithms) and is especially lacking in the areas of memory management and file systems. The expense of the volume and the lack of much hard technical information makes it a volume which is suitable for (and probably will only be used by) managers and others concerned with operating systems at a high level. For that purpose it is probably as good a volume as can be put together in a short time.

Now some detailed comments: the analysis section is really not analysis at all since much of it consists of author's explanations and advertisements for their presentations later in the volume. Most of these would be better placed at the end of their respective papers. Issues are raised and discussed, but sometimes only through the medium of text juxtaposition by the editor. Some questions are raised but not answered. This section does serve to give one a glib introduction to the rest of the volume and provides a number of pointers. While the editing is rather skillful, a true analysis of the material in the rest of the volume would be more useful.

The presentations, by and large, are rather good. They are mostly overviews and introductions, are not too technical, and are presented at a consistent level both within and between papers. In most cases the organization is rather good and easy to follow.

The invited papers are more detailed for the most part. Most present details on particular systems rather than investigating particular topic areas, such as scheduling or memory management. They are not as good as the presentations, although some of them are certainly worthwhile.

The fact that all of the attendees at the symposium and the majority of the authors and lecturers were European and British reduces the usefulness to an American audience a little. In particular, the references to systems and machines familiar to those present but not to Americans makes some of the detailed comments less than useful on this side of the Atlantic.

The index is very thin, only 1 ½ pages with generally only one reference for each item. Otherwise, the volume as a whole is a well-organized document. There is a section detailing editorial conventions and how best to use the volume.

In summary, this volume is probably useful to high-level technical managers and other decision makers who wish to obtain a broad overview of the current architectural state of the art of operating systems. It will not be very useful to individuals who wish to survey the current detailed technological state of the art.

PETER FREEMAN
Information and Computer Science
University of California
Irvine, Calif. 92664


This excellent book is the sixth in a series of volumes stemming from symposia in areas of current interest in computer science by the Courant Institute of Mathematical Sciences of New York University. It is a discursive survey of general principles confronting database system design, primarily from the stated viewpoint of the yet open (partially unsolved) problems rather than a review of the solved problems. This treatment of the subject matter is not an omission or failing—for the space of open problems is large. Its major contribution is the presentation of a wealth of ideas of what's missing and where to start in order to expand our knowledge of database design.

Early, Jacob Schwartz points out that the current problems in the database arena are related to problems of efficiency rather than problems of description. However, later chapters discuss the principal issues and properties of languages for dealing with data structures. In the opening chapter Schwartz points out that we are witnessing the emergence of a new branch of computer science, namely file theory, which ultimately could reach the significance of that of language theory—and this book is a testimonial to his prediction. In so doing, it outlines strategies for the (scientific) study of information systems, their theoretical basis, database management systems, and much more. It contributes much for assisting the reader in understanding the current problems and the present state-of-the-art of database systems.

It is this reviewer's opinion that many students, computer scientists, data-base programmers, and users will find that this book is a