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 Feiby 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 LP70 as a Language for Systems Implementation," by R. A. 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 just as position 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 1½ 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 problems confronting data base 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 data base design.

Early, Jacob Schwartz points out that the current problems in the data base area 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, data base management systems, and much more. It contributes much for assisting the reader in understanding the current problems and the present state-of-heart of data base systems.

It is this reviewer's opinion that many students, computer scientists, data-base programmers, and users will find that this book is a
necessary volume for their library—further, this book has much to offer for the designers and architects of future computer systems as well as data-base system designers.

E. C. JOSEPH
Sperry Univac
St. Paul, Minn. 55165


This book consists mainly of the descriptions of a large number of algorithms for solving mathematical programming problems. The authors state in the preface that "Because of the fast growth of the theory and applications of optimization in the past decade, there has been a need for critical review in this area, which would serve to at least two purposes: to be a manual for practitioners and an orderly description of the state of the art." The book does serve these two purposes quite well, and especially the first of these.

In giving a critical review of optimization algorithms there are two main approaches that have been taken. The first is to analyze the algorithms mathematically, giving such relevant information as the rate of convergence (or, indeed, proving convergence) and the number of function evaluations, etc., required to perform one step of the algorithm. The problem with this approach is that only a few of the algorithms of current interest admit to such precise analysis. A second approach is to perform a large number of numerical experiments with the algorithms in question, and to draw conclusions concerning the relative efficiencies of these algorithms, based on the outcome of these experiments. This is the approach taken in the present book. But the authors do not rely solely on their own experience; they make references to a wide range of experiments conducted by other researchers.

Not all of the algorithms discussed in this book have been well enough tested to allow firm statements about their performances to be made. This point is made by the authors in a statement on the bottom of p. 22. "The reader will find, unfortunately, that more research and numerical experimentation are needed to resolve some of the questions associated with the evaluation of the efficiency of iterative techniques." This is emphasized again on p. 226 in describing the results of a set of tests performed by Colville on methods for solving constrained optimization problems. The authors state, "Colville noticed that some of the methods showed a bad average performance, but were quite efficient with respect to one or more specific problems. Therefore, it is undesirable to draw any firm conclusions concerning the numerical efficiency of any optimization method from just a few numerical cases." So in reading the book, one must not expect conclusive results concerning all the algorithms discussed. However, some computational experience is referenced for each algorithm, and hopefully there is something to be learned, even from a small number of experiments.

There is an attempt to end each chapter in the book with a table giving the computer libraries which have computer programs for implementing the algorithms described in the chapter. The table contains other information such as the types of machines the programs run on, the storage requirements, and the types of problems the programs have been tested on. The authors also attempt to present the results of their own experiments with the various algorithms. They use three or four standard test problems that appear in nearly all textbooks on mathematical programming. The reporting is far more complete in some cases than in others. For example, it appears that the problem of optimization along a line, which is discussed in Chapter 3, has received so little attention that no table for it is required. Instead references to a few published experiments are given in a short paragraph. The methods discussed in Chapters 4–7 seem to have been given considerably more attention by the people working in these various areas. In particular, the conjugate gradient and variable metric methods for unconstrained optimization, which are discussed in Chapter 5, seem to have been studied quite extensively and the authors are able to report on a wide range of experiments involving these methods. These methods also submit to a fair amount of mathematical analysis and one can expound their virtues with a large amount of confidence.

Chapter 6 discusses the use of penalty functions to convert constrained optimization problems to a sequence of unconstrained ones. The reporting centers around the work of Fiacco and McCormick and their work with their SUMT (sequential unconstrained minimization technique) code. Again here is a case where we encounter a class of methods which submit to mathematical analysis and which have been thoroughly tested numerically. With the exception of a few points, which seem minor, the behavior of these algorithms is fairly well understood and the authors do a good job of reporting on this. The minor points which do not seem clear involve the selection of certain parameters in these algorithms. There are no prescriptions for choosing the parameters "optimally" and the reader is left to rely on the recommendations of people who have had considerable experience with the methods. This is not a point of criticism since certainly the reader is thereby given the best advice available.

Chapter 7 deals with constrained optimization techniques. The main techniques discussed are cutting plane methods, the method of approximate programming, feasible directions, gradient projections, and the reduced gradient method. Gradient methods for unconstrained minimization were discussed in Chapter 4.

The chapters described so far constitute the heart of the book. The first two chapters are mainly introductory. The book ends with an appendix which contains a description of the simplex algorithm for solving linear programs, and a discussion of numerical differentiation techniques. Knowledge of these is assumed in places where they are needed, such as in the discussion of the cutting plane method. Finally, in summary, the book is as the authors wished, a useful manual for practitioners. It collects together a large body of experimental knowledge on a number of algorithms and this would probably be useful to anyone who would like to obtain a program, or write a program for solving a specific problem. The book is very unsuitable as a textbook in that most of the algorithms discussed in it are merely described with no motivation given for why they might work, and in general very little theoretical analysis. Some important methods, such as decomposition techniques, are not described at all, but are mentioned and the reader referred to other sources for details.

EAL R. BARNES
IBM T. J. Watson Res. Center
Yorktown Heights, N. Y. 10598


This book evolved from notes for a seminar with the same title which has been given repeatedly at the Institute for Advanced Technology, a directorate of CDC. It intends to present an integrated guideline for the design of on-line computer systems, focusing on technical (rather than managerial) aspects, although it may help the manager to maintain a fundamental understanding of what his programmers and analysts are up to. Its six parts are meant to be basically independent, and the following is an overview of their contents.

Part I (75 pp.), Introductory Concepts, is addressed to those who may be approaching the subject for the first time. Systems are categorized according to application such as business, process control, etc. or architecture such as simplex, shared-file, multiprocessing, etc. The term On-Line System (OLS) is essentially understood as nonbatch system, as a system accepting input directly from where it is created and returning output directly to where it is needed. Hardware features for OLS are introduced with memory protection, relocatability, privileged mode, and interrupt facilities as examples of necessary features, and paging, virtual memory, microprogramming as examples of desirable features. A discussion of factors in choosing terminals and disk devices concludes this part. Although the book was published in 1972, the newer 100 million character/pack disk device introduced to the market in 1970 has unfortunately not found its way into the book yet.