mented concepts and future long-range developments. The organization is well thought out and suits the author's purposes admirably. The first chapter is devoted to defining what is meant by a programming language, to pros and cons of higher-level languages, to classifying and categorizing languages, and to a presentation of factors influencing the choice of a language. Omitted from consideration are most European programming languages, and certain languages that are not, by the author's definition, higher-level programming languages, e.g., Report Generator, Autocoder, APL and SLIP. The second and third chapters are devoted to describing functional and technical characteristics of programming languages, partly with the aim of developing a format for the later discussions of specific languages. Functional characteristics, roughly, are the historical, political, economic, and pragmatic aspects of a language, i.e., those that are not part of its definition. Technical characteristics have to do with the actual syntax and semantics of the language. This outline of discussion is spelled out in some detail and generally adhered to in the sequel. The fact that approximately equal amounts of space are devoted to functional and technical characteristics is an indicator of the tone of the discussion. The author does not attempt to teach the reader how to program in any of the languages discussed, and indeed, one certainly would not learn programming from *this* book.

As a reference work for background on specific programming languages, the book is a success. As a textbook on programming languages as a field of study, it is not. The tone of the introductory chapters is too platitudinous, and the technical information, both in depth and in organization, is inadequate to the task of imparting a feeling for what programming languages are all about. The difficulty is that this feeling can ultimately be achieved only by actually writing programs in various languages and thus coming to appreciate their fine points. Thus, the uniformity of organization across languages is achieved at the cost of omitting really deep examination of a few of them. The chapter on technical characteristics attempts to raise the issues of what choices a language designer must make, but the question of how, historically, they have been made is scattered among the different language discussions and never really explored in depth. Questions such as scoping rules, extensibility, and the structuring of data aggregates are never treated in a unified way, and the material on minor languages and on functional characteristics tends to deemphasize the importance of the technical characteristics of the major languages. There are no exercises in the book, and, indeed, the material provides no basis for exercises.

In summary: *Programming Languages*: *History and Fundamentals* is a fine source for factual knowledge, but a poor source for gaining understanding.

PAUL ABRAHAMS

Courant Institute of Mathematical Sciences
New York University
New York, New York 10012

48[12].—M. V. WILKES, *Time-Sharing Computer Systems*, American Elsevier Publishing Co., Inc., New York, 1968, iv + 102 pp., 22 cm. Price $4.95.

This is an excellent introduction to the design of time-sharing systems, which expresses concisely a number of current ideas. The text is clearly influenced by the

pioneering work done at Massachusetts Institute of Technology and the time-sharing system implemented on the Atlas at Cambridge University. A number of addressing schemes, including the paging and segmentation features provided on the GE 645 and IBM 360/67, are discussed, as well as the interesting work, on the generalized notion of a capability, of Dennis and Van Horn, and Yngve and Fabry.

The serious student will undoubtedly wish to refer to the references cited for more details.

MALCOLM HARRISON

Courant Institute of Mathematical Sciences
New York University
New York, New York 10012