Northern Ireland, 1971; and *Operating Systems Techniques*, pp. 61-71, edited by C. A. R. Hoare and R. H. Perrot, Academic Press, New York, 1973.
HANSEN, P. B. (1972a). A comparison of two synchronizing Concepts, *Acta Informatica*, Vol. 1, pp. 190-199.
HANSEN, P. B. (1972b). Structured Multiprogramming, *CACM*, July 1972, Vol. 15, No. 7, pp. 574-578.
HANSEN, P. B. (1973). Concurrent Programming Concepts, *Computing Surveys*, December 1973, Vol. 5, No. 4, pp. 223-245.
COURTOIS, P. J., and HEYMANS, F. (1973). *Information Streams sharing a finite Buffer*; *another Implementation*, MBLE, Technical Note N85, January 1973.
COOPRIDER, L. W., COURTOIS, P. J., HEYMANS, F., and PARNAS, D. L. (1973). Information Streams sharing a finite Buffer: other Solutions, *Information Processing Letters*, Vol. 3, No. 1, July 1973.
HABERMANN, A. N. (1972a). Synchronization of communicating Processes, *CACM*, March 1972, Vol. 15, No. 3, pp. 171-176.
HABERMANN, A. N. (1972b). *On a Solution and a Generalization of the Cigarette Smokers' problem*, Carnegie-Mellon University Report, August 1972.
CAMPBELL, R. H., and HABERMANN, A. N (1974). *The Specification of Processes Synchronization by Path Expressions*, University of Newcastle Upon Tyne technical report no 55, January 1974.
HOARE, C. A. R. (1974). Monitors: An Operating System Structuring Concept, *CACM*, October 1974, Vol. 17, No. 10, pp. 549-557.

Book reviews

Structured Computer Organisation by A. S. Tanenbaum, 1976; 443 pages. (*Prentice Hall*, $18.50).

This text provides an admirably mature and illuminating study of computer architecture. The subject matter is uniformly well presented, but the feature of most significance is the overall approach to the subject. All too frequently, texts on computer architecture distinguish a level of architecture broadly equated to the demarcation between hardware and software, and concentrate on the implementation of this level, principally by hardware components. This text distinguishes a number of levels, each of which may be supported by, and may itself support, both hardware and software components. The result is a logical exposition of architecture viewed at different levels.

The particular levels on which attention is focussed are divided into two classes, namely, those which are implemented by interpretation and those implemented by translation. In the former category are the microprogramming level, at which hardware interprets sequences of microprogram steps, the conventional machine level, at which the microprogram interprets sequences of basic instructions, and the operating system level, where instructions are interpreted either by microprogram or by the operating system. Levels implemented by translation are typified by the assembly language level; the text does not extend the treatment to a consideration of translated high level source languages, but does discuss architectural features required to implement an interpreted machine level which would support the execution of high level source language statements. A final chapter considers various methods for implementing multi-level machines, including a most illuminating study of 'self virtualising machines', using the IBM VM/370 system as an illustration.

The major examples used throughout the text for in-depth study are the IBM System/360 and System/370, the CDC 6000 and Cyber series, and the DEC PDP-11 series. These provide well-varied illustrations of the subject matter, and are supplemented by additional examples in particular subject areas (e.g. Burroughs B1700 is also used as an example when discussing the microprogramming level). From the viewpoint of a reader in the United Kingdom, it is unfortunate that the illustrations are not extended to include, for example, the ICL 1900 architecture, which provides a further excellent illustration of the various levels. In particular, the composition of the conventional machine level, which can be broadly equated to the ICL 1900 Executive level, provides an excellent contrast with the examples quoted in the text.

It is a tribute to the author's presentation of the material that the reader will regret that the book is not further extended to consider additional levels. The hardware level of interpretation is only touched upon (the reader should beware lest he assume that microprogram interpretation of the conventional machine level is uniformly applied) and operating system structure is not considered in depth; the latter could well be described by means of an interpreted level of machine supporting operating system activities. However, these omissions by no means impact on the stimulating treatment given of structured computer organisation, and the examples and bibliography support the text material in a well chosen manner.

This text is to be recommended to all students of current-day computing system architecture.

D. J. HOWARTH (London)

Mathematical Programming second edition, by C. McMillan, 1975; 650 pages. (*John Wiley & Sons Inc*, £9·75).

When I attended an international mathematical programming symposium in London in 1964 I was surprised at the variety of useful extensions of linear programming methods for nonlinear problems that depend much more on intuition and experience than on mathematical analysis. Since then many good mathematicians have studied nonlinear programming and consequently some techniques have been developed that are often much more powerful than the earlier methods. However, due probably to the fact that the first edition was published in 1970, this book reminds me much more of the London symposium than of recent work. It seems to be written for people who are ignorant of mathematics.

Therefore the few mathematical ideas that occur are introduced in a very detailed manner. For instance, even the definition of a derivative is explained by reference to the speed of a car, and the elementary properties of vectors are stated before they are used. The most advanced method that is given for unconstrained optimisation is the method of steepest ascent. The terms 'positive definite' and 'conjugate' are not mentioned.

Instead the language of the book is that the ideas are developed through examples which are mainly drawn from highly simplified business and ordinary life situations. For instance linear programming is introduced by a scheduling problem where two machines can each manufacture two products. It is fascinating to read how well a subject can be presented through examples, but some people will be irritated by the verbosity and occasional lack of exactitude. The examples account for about half of the length of the book. Most of the remainder is descriptive and there are exercises for the reader at the end of each chapter.

The chapter headings give a fair survey of contents and they are as follows: Fundamental concepts in linear programming, Linear programming extensions, Assignment and distribution methods of linear programming, Classical optimisation, Gradient methods of optimisation, Simplex based methods, Geometric programming, Dynamic programming, The branch and bound algorithm, Integer linear programming, Zero-one programming, Applications of zero-one programming, Network optimisation and Mathematical programming with multiple objectives. I recommend this book as an elementary introduction to all of these subjects because its style makes it easier than usual to understand.

M. J. D. POWELL (Cambridge)