Listed are abstracts from recent papers and books by IBM authors. Inquires should be directed to the publications and publishers cited.

Analysis and performance of inverted data base structures, A. F. Cardenas (RES San Jose, CA), *Communications of the ACM* **18,** No. 5, 253–263 (May 1975). The architecture of a data base system may be structured into several hierarchical levels but, in actual practice, these distinct levels have not been generally envisioned. The inverted data base (file) organization is analyzed, considering implementation-oriented aspects. It is shown that the potential advantage of inversion may be significantly impaired if the directory is organized as a simple sequential file. Consequently, the directory is viewed as another large data base which itself is subjected to inversion.

Automatic structuring of programs, G. Urschler (SPD Vienna, Austria), *IBM Journal of Research and Development* **19,** No. 2, 181–194 (March 1975). A method is described that allows the translation of a traditionally written (unstructured) program into a set of top-down structured, semantically founded, GOTO-free modules. The method reveals not only the logic of a given program in a most natural way, but it also reduces code duplication to a minimum. It is further shown how the obtained structured program can be mapped back into a GOTO program in such a way that all GOTOs are backwards branches and their number is minimal. The connection between recursively and iteratively structured programs is demonstrated using the WHILE, DO FOREVER, and multilevel EXIT statements. Extensions of the method show the structuring of source programs containing block structures and subroutines.

Abstracts

Computer data-base organization, James Martin (Systems Research Institute, New York, NY), Prentice-Hall, Inc., Englewood Cliffs, New Jersey (1975). This book, an introductory text, is designed to familiarize the reader with the many alternatives possible in data organization and with the trade-offs between them. Part one, logical organization, describes the data base as the programmer sees it. Included are chapters on the objectives of data base systems, the CODASYL Data Description Language, IBM's Data Language/1, relational data bases, and varieties of data independence. Part two, physical organization, describes the physical placement of data on electromechanical storage units. Included are chapters on pointers, indexed sequential organizations, physical representations of tree and plex structures, data compaction, virtual memory and storage hierarchies, inverted file systems, and associative memory. 558 pp.

A cost-oriented algorithm for data set allocation in storage hierarchies, V. Y. Lum (RES San Jose, CA), M. E. Senko (RES Yorktown Hts., NY), C. P. Wang (ASDD Mohansic, NY), and H. Ling (DPPG Harrison, NY), *Communications of the ACM* **18,** No. 6, 318–322 (June 1975). In systems with multilevel storage hierarchies, data set allocation is usually based on qualitative, ad hoc decisions. Here an algorithm for optimal allocation of data sets is outlined, considering data staging, migration, and dynamic allocation techniques. Many significant parameters of data set usage, including storage space cost, CPU operating cost, and the cost of moving the data set from one storage level to another are considered. The parameters used in the algorithm have been defined so that their values can be easily gathered by the computer system, although initially the owners of the data sets will have to provide an estimate of the parameter values.

Performance evaluation of computer systems, R. Ashany, Ed., *IBM Journal of Research and Development* **19,** No. 3 (May 1975). (G322-0082). Performance evaluation is a process consisting of a sequence of states, such as problem formulation, parameter selection, model construction, model tuning, result validation, and result interpretation. The first seven papers in this topical issue describe the application of various techniques to the performance evaluation and optimization of computer systems. The subsequent four papers report on the development of innovative techniques that make some existing models more applicable to a wider range of problems.

The implicit conclusion of these papers is that, in addition to the application of sophisticated techniques from the fields of mathematics, statistics, operations research, information theory, optimization, and other related disciplines, a good understanding of the interactions among hardware, software, firmware, and load profiles in general, and of the operating system in particular, is an absolute must for the creation of realistic and efficient models.

The structure and design of programming languages, J. E. Nichols (IBM UK Laboratories), Addison-Wesley Publishing Co., Reading, Massachusetts (1975). The design of high-level procedural programming languages is the concern of this book. Programming languages now in widespread use contain many common facilities, although these are made available to the user in a variety of different ways. The approach taken in this book is to present those features that form an essential part of any high-level language, such as scalar and aggregate variables, arithmetic and logical expressions, and iteration elements. For each of these, a summary of underlying elements is stated, followed by a description of how the feature is represented in various high-level languages.

This volume, the third in *The Systems Programming Series,* is intended for applications and systems programmers who use high-level languages in their day-to-day work. 572 pp. Bibliography.