

Listed are abstracts from recent papers by IBM authors. Inquiries should be directed to the publications cited.

## Abstracts

**An analysis of page allocation strategies for multiprogramming systems with virtual memory**, D. D. Chamberlin, S. H. Fuller and L. Y. Liu, *IBM Journal of Research and Development* 17, No. 5, 404-412 (September 1973). In a multiprogramming, virtual-memory computing system, many processes compete for the main storage page frames and CPUs of the real system. It is customary to define a subset of these processes called the "multiprogramming set" (MPS), and to allocate resources only to those processes currently in the MPS. Each process remains in the MPS for a limited time and is then demoted. The system paging manager controls the size of the MPS; it allocates the available page frames among the processes in the MPS and fetches appropriate pages into the page frames. A model is then described that assumes the most critical resources of the system to be page frames and the paging channel (i.e., there is no significant CPU contention). The model makes certain assumptions about the page fault rate of processes as a function of page frames allocated, and about the page fetch time as a function of mean load on the paging channel. The model also incorporates a definition of the value of a given page allocation in terms of system throughput. The model is used to study various strategies for choosing an MPS and allocating page frames among processes. For simple cases, the model yields an exact optimal strategy. A heuristic strategy is proposed for dealing with more complex cases, and is shown by the model to be reasonably near optimal. The heuristic strategy monitors the page fault rate of each process and chooses an allocation such that each process can be executed at a reasonable rate, while ensuring that the paging channel is neither overloaded nor underloaded.

**Characterization of program paging in a time-sharing environment**, Y. Bard, *IBM Journal of Research and Development* 17, No. 5, 387-393 (September 1973). This paper describes a method for predicting the paging behavior of a program in a virtual memory multiprogramming environment. The effect of overall system activity on the program is summarized in one parameter, the page survival index. The model correlates well with observations taken on programs running under CP-67. The model can be used for paging load prediction, simulator input verification, and evaluation of program rearrangement and sharing.

**Cyclic queues with bulk arrivals**, I. Adiri, *Journal of the Association for Computing Machinery* 20, No. 3, 416-428 (July 1973). This paper deals with a single-server station (a computer) where each customer's demand comprises an independent random number of jobs (programs). Under certain assumptions, two cyclic disciplines are mathematically analyzed: (a) continuous job service—a round-robin discipline where the quantum's length is distributed as the service requirement of a job; (b) intermittent job service—a double round-robin discipline—in the first instance in terms of the jobs within the customer's demand, and in the second in terms of the customer himself.

**The design of APL**, A. D. Falkoff and K. E. Iverson, *IBM Journal of Research and Development* 17, No. 4, 324-334 (July 1973). This paper discusses the development of APL, emphasizing and illustrating the principles underlying its

design. The principle of simplicity appears most strongly in the minimization of rules governing the behavior of APL objects, while the principle of practicality is served by the design process itself, which relies heavily on experimentation. The paper gives the rationale for many specific design choices, including the necessary adjuncts for system management.

**Locality in page reference strings**, G. S. Shedler and C. Tung, *SIAM Journal on Computers* 1, No. 3, 218-241 (September 1972). A probabilistic model is presented of program material in a paging machine. The sequences of page references in the model are associated with certain sequences of LRU stack distances and have reference patterns formalizing a notion of "locality" of reference. Values for parameters of the model can be chosen to make the page-exception characteristics of the generated sequences of page references consistent with those of actual program traces. The statistical properties of the execution intervals (times between page-exception) for sequences of references in the model are derived, and an application of these results is made to a queuing analysis of a simple multiprogrammed paging system. Some numerical results pertaining to the program model and the queuing analysis are given.

**Modeling of storage properties of higher-level languages**, K. Walk, *International Journal of Computer and Information Sciences* 2, No. 1, 1-24 (1973). The role of storage in the characterization of higher-level programming languages is discussed. Assignment, in particular, has significantly different meaning in different languages, which can hardly be understood without reference to an underlying model of storage. A general storage model is sketched which can be specialized to a model of ALGOL68 or of PL/I storage. The same model is used to discuss language features allowing highly flexible data structures.

**On the precision attainable with various floating-point number systems**, R. P. Brent, *IEEE Transactions on Computers* C-22, No. 6, 601-607 (June 1973). For scientific computations on a digital computer the set of real numbers is usually approximated by a finite set  $F$  of "floating-point" numbers. We compare the numerical accuracy possible with different choices of  $F$  having approximately the same range and requiring the same word length. In particular, we compare different choices of base (or radix) in the usual floating-point systems. The emphasis is on the choice of  $F$ , not on the details of the number representation or the arithmetic, but both rounded and truncated arithmetic are considered. Theoretical results are given, and some simulations of typical floating-point computations (forming sums, solving systems of linear equations, finding eigenvalues) are described. If the leading fraction bit of a normalized base-2 number is not stored explicitly (saving a bit), and the criterion is to minimize the mean square roundoff error, then base 2 is best. If unnormalized numbers are allowed, so the first bit must be stored explicitly, then base 4 (or sometimes base 8) is the best of the usual systems.

**Placement of records on a secondary storage device to minimize access time**, D. D. Grossman and H. F. Silverman, *Journal of the Association for Computing Machinery* 20, No. 3, 429-438 (July 1973). The problem considered is how to place records on a secondary storage device to minimize average retrieval time, based on a knowledge of the probability for accessing the records. Theorems are presented for two limiting cases. A numerical example for an intermediate case is also given.

**Processor utilization in multiprogramming systems via diffusion approximations,** D. P. Gaver and G. S. Shedler, *Operations Research* 21, No. 2, 569-576 (March-April 1973). Cyclic queuing systems have been proposed by several authors in the study of the behavior of multiprogrammed computer systems. Programs in the system wait for service at the central processor unit (CPU); then, after page fault or input-output request at a data transmission unit (DTU), the process repeats until the program completes. Semi-Markov analysis of such systems, based on the apparently plausible assumption of independently but exponentially distributed CPU burst time, and independent, but nearly constant DTU time may be conducted. This paper presents some very simple approximations based on a continuous-state approximation—the simple diffusion with two reflecting barriers—to describe the CPU utilization. Computational experience from which the quality of the approximations can be assessed is reported.

**Response time characterization of an information retrieval system,** H. F. Silverman and P. C. Yue, *IBM Journal of Research and Development* 17, No. 5, 394-403 (September 1973). A methodology for computer performance evaluation based on the statistical characterization of response time is described. The results of its application to an information retrieval system are presented. The first part of the paper gives a general discussion of measurement techniques, data reduction procedures and the structure of the system being examined. A set of "system environment" parameters and a set of "job" parameters are then defined and appraised in terms of actual measurements collected over two different weekly periods. Various ways of using the statistical characterization for improving performance are then considered.

**System formulation and APL shared variables,** R. H. Lathwell, *IBM Journal of Research and Development* 17, No. 4, 353-359 (July 1973). The problem of providing communication with APL programs was approached by formulating systems as collections of autonomous processors communicating on interfaces consisting of shared variables. This paper discusses the formulation of a theoretical APL system and cites experience with a prototype APL shared variable system which both uses and provides shared variable interfaces.