HONEYWELL

SOFTWARE BULLETIN

SERIES 200

SERIES 200/OPERATING SYSTEM-MOD. 1 (MASS STORAGE RESIDENT)

The Series 200/Operating System — Mod 1 (Mass Storage Resident) is a computer-management system designed for medium-scale installations having a mass storage device (for example, a Honeywell Mass Memory File) and control unit and from 8,192 to 65,536 characters of main memory. Incorporated into the system are supervision, data management, program development, and service functions. Routines are selected and combined to meet each user's specific data processing requirements and to exploit systematically the capabilities of his equipment configuration. Because the program runs with only the necessary routines and equipment, the user is not faced with an uneconomical fixed overhead.

The basic Mass Storage Resident (MSR) Operating System equipment configuration is 8,192 characters of main memory, one mass storage device and control unit, a card reader and a card punch (or a card reader/punch), a high-speed printer, and the Advanced Programming Instructions. However, some programs do not require all of the preceding equipment, while others require more than 8,192 characters of main memory. The system also permits the use of magnetic tape, punched card, paper tape, and communication devices. The design of the software modules is such that effective use is made of additional main memory, mass storage, and peripheral equipment when it is available. These modules are linked in a common structure of data management conventions and operating procedures and are controlled by a common supervisor. Each individual module, however, is capable of performing its own task in an independent manner.

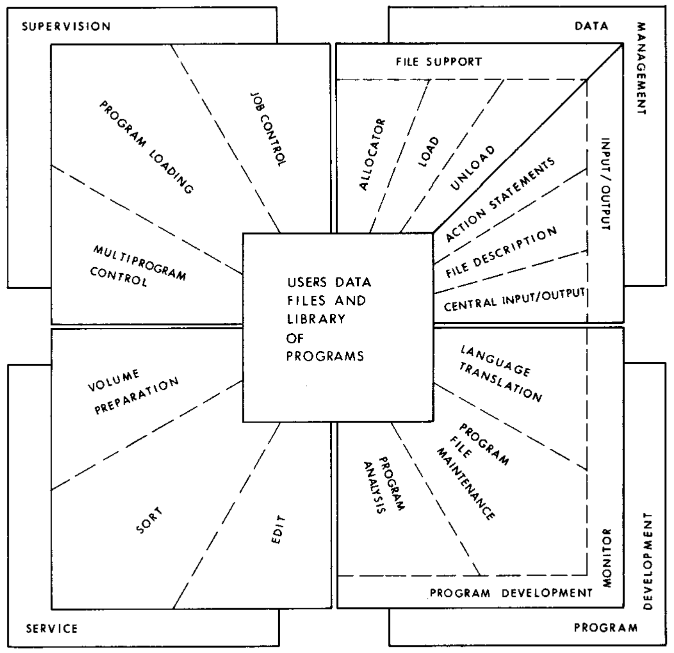
OPERATING SYSTEM

The MSR Operating System is a comprehensive package of integrated software modules which are designed to assist the user in making the most efficient use of his mass storage equipment. The four major functions — supervision, data management, program development, and service — embrace the various routines that are described below.

Supervision

All operations in the MSR Operating System are performed under the general control of a supervisor program. Portions of this program are permanently resident in main memory, where all but the communications segment have a floating status, while other portions are loaded from mass storage as required.

Specifications remain subject to change in order to allow the introduction of design improvements.



The supervision function performs job control, program loading, and multiprogram control. Job control effects automatic transition from one job to the next. Program loading encompasses all facets of program segment loading, including locating the desired segment in the program file upon request. Multiprogram control allocates the processing time of the central processor in such a way that a peripherally oriented program can be run simultaneously with a data processing program.

Data Management

Data management is concerned with the control of and access to files. Two types of functions are included — file support and input/output. The file support functions create, organize, and reorganize files in addition to converting these files from one storage medium to another. The input/output functions retrieve and write data within the files.

File Support: File support uses three basic types of file organization:

1. Sequential — in which the data is organized for sequential access,
2. Direct access — in which the data is organized for random access, and
3. Indexed sequential — in which the data is organized for both sequential and random access.

(Continued on reverse side)

Honeywell

ELECTRONIC DATA PROCESSING

121.0005.141B.0-407
8566
Printed in U.S.A.

When ordering this publication please specify Title and Underscored portion of File Number.

File support includes allocator, load, and unload routines. The allocator routine reserves an area of a volume (for example, a Mass Memory File Pack) for a file as described by the user, modifies the volume directory as necessary, and formats and initializes the area if necessary. The load routine loads data into a previously allocated file, establishes the sequence of items, and sets up indexes as required by the file organization. The load routine can accept input from one of several different input media. The unload routine unloads data from a mass storage file, reorganizing the data so that storage utilization and access time are improved when the file is reloaded. The unload routine can produce output on one of several different output media.

Input/Output: The input/output functions equip the programmer with macro routines to access files that are arranged in one of the three supported file organizations. Three types of macro routines — action statements, file description, and central input/output — are provided. The programmer selects the appropriate macros from each type, depending on the access mode desired, the storage devices used, and the number and organization of his files.

Action statements name the action the programmer wants performed and the file to which it is addressed. The file description macros describe the structure of the program's files. There must be one file description macro for every file in the program. The central input/output routines perform the common logical input/output operations for all files, control the sharing of physical resources among the peripheral devices and handle physical input/output operations.

Program Development

Program development includes routines for language translation, program file maintenance, and program analysis. A monitor within this function controls the automatic sequencing of the various steps in a program development job.

The language translators include an Easycoder assembler, a COBOL compiler, and a Fortran compiler. The symbolic language of the assembler is comparable to Easycoder D in the Series 200/Operating System — Mod 1 (Tape Resident); the COBOL language is comparable to COBOL B in the Series 200/Basic Programming System; and the Fortran language is comparable to Fortran D in the Tape Resident Operating System.

Program file maintenance employs routines to maintain either source- or machine-language files on mass storage. The analyzer produces a symbolic cross-referenced listing for an Easycoder symbolic program. Main memory and mass storage dump routines are provided.

Service

The service function performs such common processing tasks as volume preparation, sorting, and editing.

The volume preparation routine prepares a mass storage volume for use in the MSR Operating System by checking for bad surface areas, formatting tracks, and establishing the volume label and directory. The mass storage sort operates with one mass storage device. The sorting key may be composed of up to ten separate fields. Key sorting is performed, and other information extracted from the item may be associated with the sorted keys. A mass storage edit routine is provided to edit and print selected areas from mass storage.

DELIVERY

Initial delivery of the basic elements of the MSR Operating System will be made with the mass storage hardware. This includes:

- Supervision
- Easycoder Assembler
- Support of sequential and direct-access files
- Machine-language program file maintenance

Later extensions will include:

- Multiprogramming capability
- COBOL Compiler
- Fortran Compiler
- Support of indexed sequential files
- Source-language program file maintenance

SUMMARY OF ADVANTAGES

- Relieves the operator of detailed and burdensome execution supervision
- Standardizes operating procedures
- Permits operation with only the needed functions and features
- Enhances the running of user-written programs, library programs, and systems programs as integral parts of the operating system
- Enables programs and files to be located and processed without time-consuming searches
- Makes optimum use of storage surface through a flexible file design that allows the user to specify his record size
- Provides execution of stacked jobs without operator intervention
- Allows tailoring and specialization of precoded, fully tested library routines to meet program needs
- Maintains systems programs simply and economically
- Assures efficient use of the central processor and peripheral units through multiprogramming