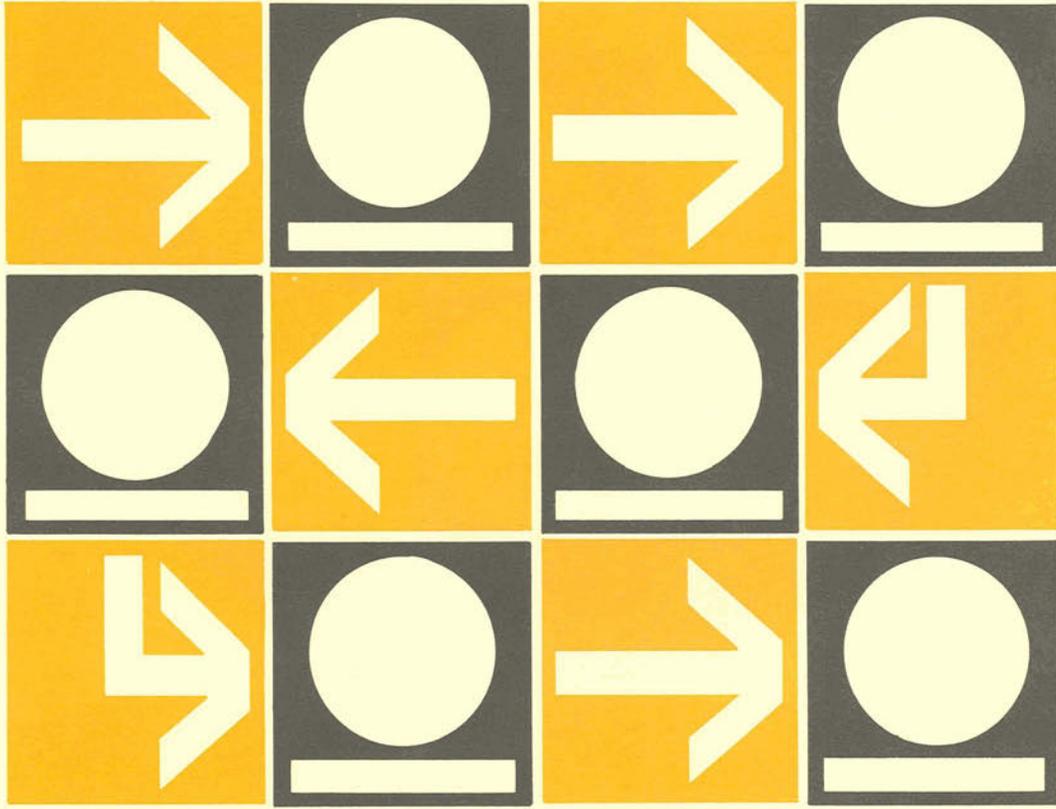
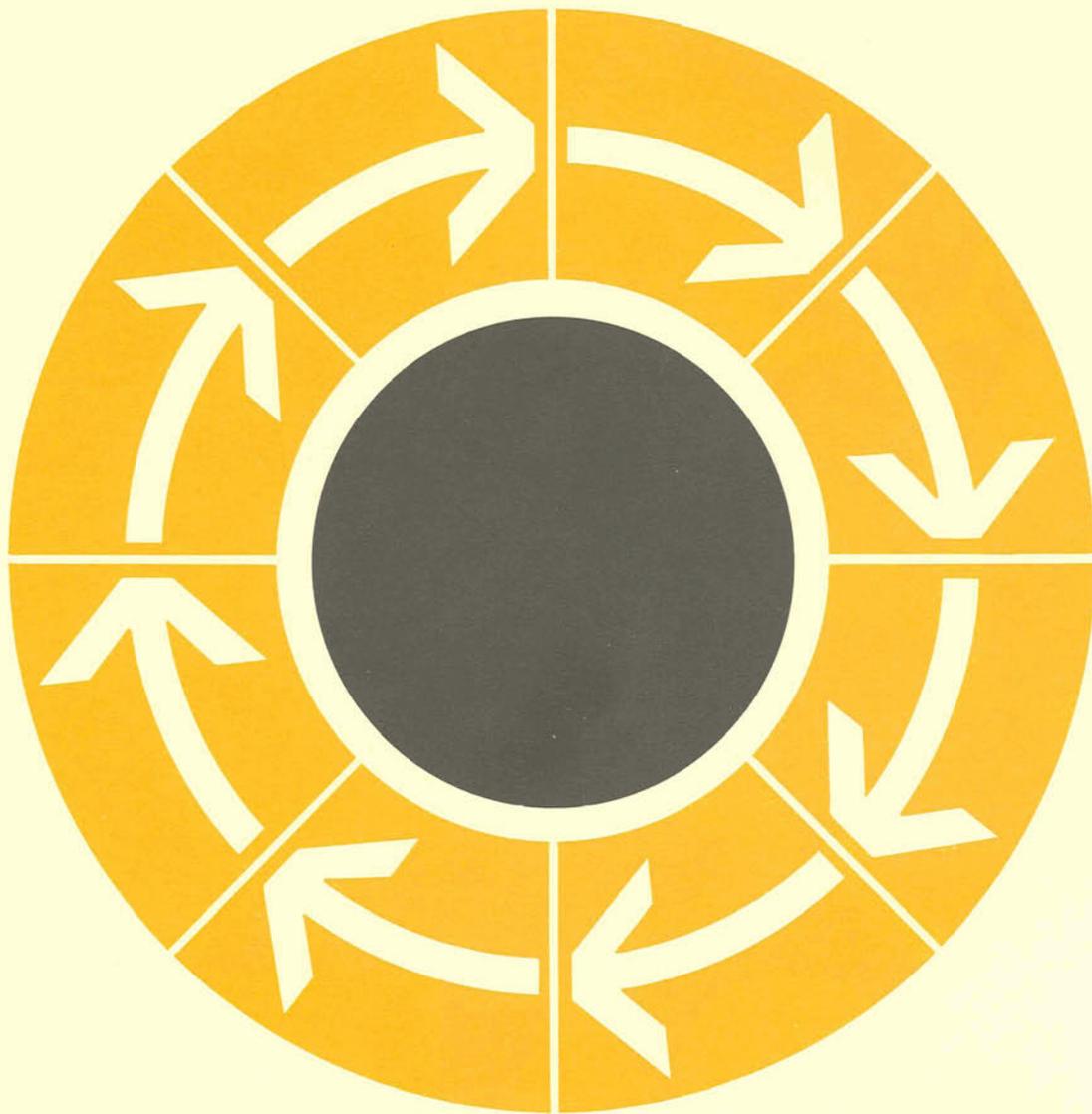


Operating System
Orientation for
Management





New-generation computers, such as Honeywell's Series 200, offer more computing power per dollar than any previously available data processing systems. It is not uncommon for members of Series 200 to operate at speeds four to six times faster than those of their second-generation counterparts. Yet in evaluating a move up to a new computer, data processing management must realize that more computing power per dollar is not the whole story. The ability to compute six times faster does not, by any stretch of the imagination, guarantee that a data processing workload can be processed in one-sixth the time. While technological advances have made significant contributions to improving computer performance, the time required to perform manual operations has remained unchanged. In effect, technology can do very little to improve manual speed. As computers get faster, the time required to perform manual operations assumes an increasingly large percentage of total processing time.

The Operating System Approach

As the name implies, operating systems have evolved from the essential need to automate manual procedures wherever possible — to delegate much of the responsibility of running the computer installation to the computer itself. Human intervention wastes a tremendous amount of processing time due to the disparity in operating speeds between the computer and its human operator. Thus, the first objective of an operating system is to automate the steps involved in job-to-job transition and in setting up for a specific job. By automating manual procedures wherever possible, or at least overlapping manual procedures with other processing when automation is not possible, an operating system acts as a programmed computer operator. It performs all of the supervisory functions necessary to handle the scheduling and processing of all jobs for which the computer is used, and it performs these

functions at electronic speeds. In terms of improving computer operating effectiveness, this means that the measure of computer idle time can be reduced from minutes to fractions of a second.

When properly designed and implemented, an operating system can play a significant role in harnessing the full potential of a data processing system. This report explores operating systems from several different viewpoints in order to describe what operating systems are, how they work, and the benefits that can be derived from them.

ON THE FACING PAGE: *Two approaches to computer control. At top left, the operating system approach maximizes throughput by providing automatic job-to-job transition for continuous nonstop operation. At bottom left, manual operations between jobs waste valuable processing time due to the tremendous disparity in operating speeds between the computer and the human operator.*

Advantages of Operating System Control

Operating systems are aimed at one major goal: improving computer operating effectiveness. They can provide a number of distinct advantages in terms of better system performance, wider application, and reduction in manpower costs. Therefore any evaluation of a new computer should include a careful analysis of how closely the services offered by the computer's operating system meet the requirements of the user's installation. Such an analysis will indicate the degree to which the user's major data processing objectives will benefit from the use of an operating system. The following descriptions indicate some of the *potential* benefits that can come from using a well designed operating system.

Efficient Program Testing

One of the first contributions that operating systems made to improved computer operating effectiveness was in the area of program testing. Operating systems provide standard program testing facilities which make it possible to perform program checkout by remote control. These testing facilities can perform sequential checkout of several stacked programs, automatically producing detailed documentation with which to evaluate the programs being tested. This approach to program testing makes it possible to avoid the time-consuming and costly technique of testing programs one at a time using the computer console as the source of program status information.

Increased Throughput

One of the best measures of the value of an operating system is its ability to maximize the throughput of a data processing system. Throughput — the total amount of work which the system can perform in a given amount of time — depends on many factors. For example, it depends on the ability of the system to process jobs in a continuous stream without interruption. An operating system provides this ability by automating the transition from one job to another. The operating system maintains a schedule of the jobs to be performed and the equipment required by each. It monitors the progress of each job so that upon completion of one job, another can be started automatically without delay. Furthermore, the operating system can schedule necessary manual intervention, such as mounting of tape reels, so that these activities are overlapped with the processing of other jobs.

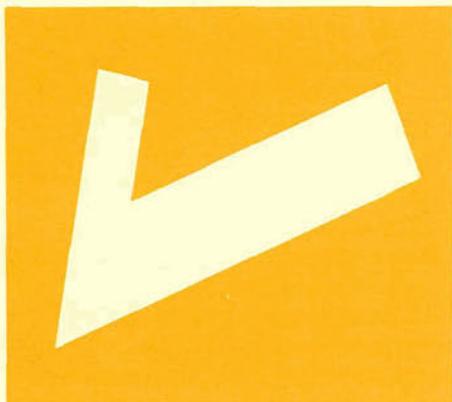
A high level of throughput also depends on the efficient utilization of the computer's resources. This can be accomplished by the use of multiprogramming, a scheme whereby several programs concurrently time-share and space-share the storage, input/output, and control facilities of the system. The operating system keeps as much of the system busy doing productive work as much of the time as possible. It determines the individual requirements of each program in terms of memory space and input/output equipment. It

determines the logical dependence relationships which may exist between programs, as well as the relative urgency of each individual job.

Reduced Turnaround Time

Turnaround time is the interval between the submittal of a job for processing and the delivery of the results. This interval has three components: productive time, during which the computer's electronic speed is applied to creating the desired results; nonproductive time, during which the computer performs input/output media-conversion operations; and idle time, during which the computer waits for an operator to perform various manual tasks.

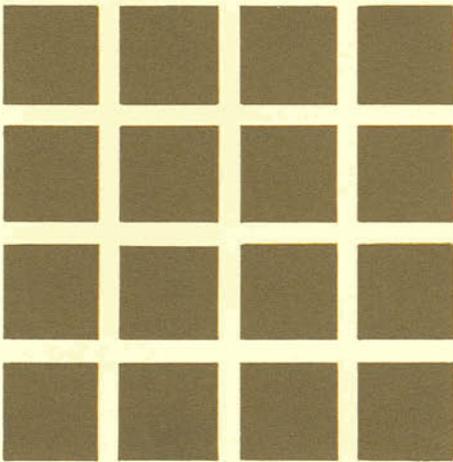
An operating system can not only enhance the utilization of the computer's resources during productive time, it can significantly reduce costly nonproductive and idle time as well. By limiting and controlling the amount of human participation in the data processing operation and by providing more direct communication between the job and the computer, an operating system can eliminate many of the delays caused by human intervention between jobs. By handling input/output media conversions, such as converting data from punched cards to magnetic tape, concurrently with other job processing, an operating system can eliminate nonproductive overhead from job turnaround time.



Standardization

An operating system is a powerful management tool because it imparts a high degree of standardization to both programming and operating procedures. Since an operating system is the intermediary through which programmer and operator communicate with the computer, it can superimpose a highly efficient and uniform communication scheme on all data processing activities. This scheme assists the programmer by providing fast and simplified access to frequently used routines and programs; facilities for efficiently storing, testing, modifying, and retrieving programs and data; and a control system that monitors and responds to status changes within the computing system.

Assistance to the operator is provided by such features as uniform and unambiguous operating instructions; automatic handling of file label checking, error detection, and recovery; and automatic logging of system performance.



Simplified Program Preparation and Maintenance

The language processing, program editing, and program maintenance functions provided by an operating system can reduce the time, training, expense, and manpower required to plan, prepare, and maintain problem programs and applications. Language processing functions are handled by language

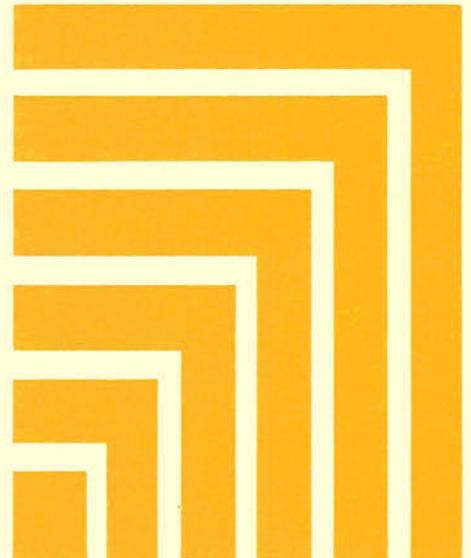
translators, that is, programs which translate the user's definition of a problem solution into a language acceptable to the computer. By offering a variety of language translators which produce object programs in a standard format, an operating system can insure that the user will be able to express the solution to a problem in the language best suited to his needs. For example, problem solutions can be expressed in the mathematical notation of Fortran, the concise business-oriented language of COBOL, or the flexible symbology of assembly language. Furthermore, portions of a problem solution can be written in different languages and then automatically combined by the operating system to form a single program.

Processes such as storing, modifying, and maintaining programs come under the heading of program editing and maintenance functions. Operating system control reduces the time and expense associated with filing, altering, and rearranging programs. Furthermore, programs can be selected from the library and ordered in a sequence best suited to the requirements of a specific job, again under automatic operating system control.



Flexible and Orderly Growth

As data processing requirements increase, an operating system can provide the basis for orderly expansion of performance and application. By providing a standard system of communication and control, the operating system makes it possible to incorporate new computing and input/output facilities without destroying compatibility with existing programs and applications. New applications can utilize existing data files and can communicate with existing programs without regard to specific changes in equipment.



Structure of an Operating System



An operating system is a framework within which all of the user's data processing jobs can be scheduled and performed. The following is a brief description of the basic components of this framework, using the Honeywell Series 200 Operating System — Mod 2

as an example. The framework provided by an operating system can be viewed in terms of three functional units: Job Control, Data Control, and Program Preparation and Maintenance. Each unit contains a number of program modules which can

be combined in a variety of configurations to meet a user's specific operating requirements. Working together as a team, these program modules provide a highly efficient means of automating the management of the computer system.

Program Preparation and Maintenance

The most familiar program preparation function is language processing. Programs written in Fortran, COBOL or assembly language are translated to machine-language program modules. These program modules are relocatable in the computer's memory and can be combined with other modules to form a variety of complete programs.

An equally important program preparation function is the creation of complete programs by selecting existing program modules, providing linkages between the modules, and assigning

actual storage addresses to each module.

Program maintenance functions include those activities associated with adding, deleting, and modifying program modules in the user's program file. Standard maintenance routines handle all the clerical operations necessary to keep programs up to date.

Job Control

The primary function of job control is to automate the steps involved in job-to-job transition. Job Control is handled by program modules called monitors. The Mod 2 Operating System uses two monitors; one resides in the computer's memory at all times, while the other is called in periodically to assist in the automatic transition process. Job Control automatically handles such procedures as collecting the output produced by the previous program, locating the next program to be executed and loading it into memory, and coordinating input/output device as-

signments. The user prepares various control cards which describe to the monitor how the transition from program to program should be accomplished.

In addition to reducing the idle time between program executions, Job Control can also dramatically reduce total job time. Simply stated, a job is a collection of related programs. To speed up the processing of a job, Job Control provides for automatic "stacked-job processing." Under stacked-job processing, any number of processing functions such as compilation, mainte-

nance, and execution may be successively applied to the same job. In other words, each job in a stack of jobs can be processed to completion before the next job is started.

A third function of Job Control is to serve as a communication link between operator and system. Job Control can provide messages to the operator advising him on the status of various operations or requesting necessary operator actions. Similarly, the operator can alter the operation of the system or request specific information by issuing commands to Job Control.

Data Control

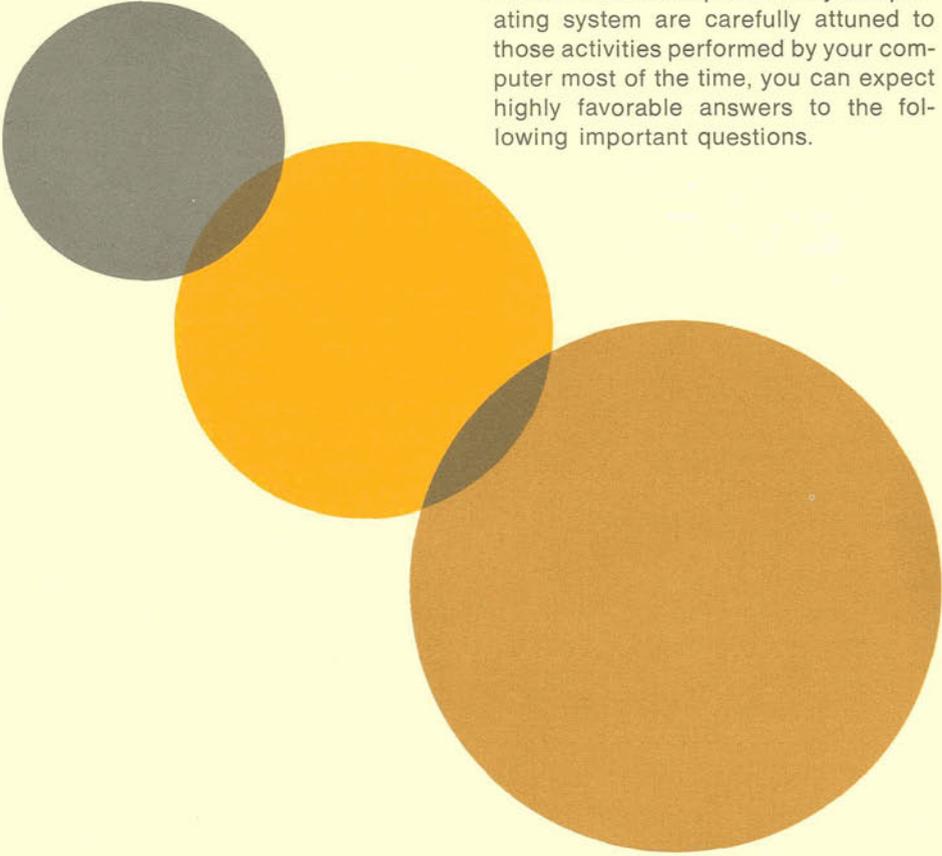
Data Control provides facilities for managing all functions related to the creation and maintenance of the data base, that is, the entire collection of information which enters or leaves the computer memory. These facilities achieve efficient storage, flow, and retrieval of all data within the system. Data Control handles the physical exchange of data between computer memory and unit record, magnetic tape, mass storage, and communication equipment. Associated with data transfer are other functions such as error detection and correction, data buffer-

ing, data blocking and unblocking, dynamic scheduling of input/output facilities, and overlapping of input/output operations with computing. Data Control performs all of these functions automatically.

In addition to managing the physical exchange of data between the computer and peripheral devices, the Data Control function also includes management of logical data files at a level which is independent of the physical characteristics of the files and their storage devices. Symbolic names can be assigned to files. The operating sys-

tem maintains a symbolic file catalog which is similar to a library catalog. When a file is requested by its symbolic classification, the operating system can consult the file catalog to determine the physical identity and location of the file so that it may be retrieved.

Some Criteria for Evaluating Operating Systems



The fact that operating systems in general are directed toward the same goal of improving computer operating efficiency should not imply that all operating systems are equally successful in attaining that goal. The design of an operating system has a direct influence on how effective it will be in automating the management of your computer. When the services provided by an operating system are carefully attuned to those activities performed by your computer most of the time, you can expect highly favorable answers to the following important questions.

How Much Will the Operating System Cost?

Although an operating system can provide several benefits to the user, it is important to take note of how much these benefits are going to cost. In order for an operating system to perform its many functions it requires space in the computer memory, certain peripheral devices, and processing time. Collectively, these requirements represent operating system overhead. In order for the user to tolerate the cost of this overhead, it is imperative that the operating system yield significant advantages which could not otherwise be attained.

Realizing that a single operating system design applied to all of the requirements of all Series 200 users would create intolerable overhead costs for those users least able to afford them, Honeywell has developed an operating system which is divided into three models. Each model is designed to fit a specific range of core memory and system environment features. This approach allows the smaller user to avoid major operating system overhead by using an operating system model more closely suited to his requirements. Similarly this approach does not limit the user of a larger system such as a Model 4200 to an operating system designed to service the needs of a Model 120.

The following table lists the equipment requirements for the various operating system models as an indication of what it costs to use them.

Operating System Model	Minimum Memory Overhead (K = 1,024)
MOD 1	1.4K char.
MOD 2	17.5K char.
MOD 8	64K char.

Is the Operating System Easy to Use and Maintain?

An operating system is supposed to simplify computer operations, not complicate them. Therefore the means of communicating with and maintaining an operating system are important considerations. The value of an operating system is diminished appreciably if communication between user and operating system is error prone due to the need for complex and lengthy parameters with which to initiate operating system functions. Also, the value of an operating system is diminished if maintenance is an overly time-consuming task.

As an example of how easy it is to maintain a Honeywell operating system, consider the time required to generate a typical business-oriented version of the Mod 2 Operating System. Unlike some operating systems which require a series of complex, time-consuming programs for initial generation, the Mod 2 Operating System uses the same single-phase component which updates the system files to create a working version of the operating system itself. System generation is both selective and efficient. A specialized operating system is tailored to the user's specific needs by incorporating only those modules necessary to service his needs. In general this can be done in less than 15 minutes. Minor changes can later be made to the operating system without having to regenerate it completely.

Can the Operating System Utilize Existing Programs?

An integral part of the Series 200 Operating System are Honeywell's renowned Liberator programs. These programs make it possible for users of IBM 1400 series equipment to translate existing program libraries automatically into fully compatible Series 200 programs. By providing what is essentially an automated reprogramming facility, Liberator programs dramatically reduce the time and expense normally associated with converting from one computer to another.

In addition to automating the conversion to Series 200 computers, the Liberator programs automatically improve the performance of converted programs by:

1. building input/output/compute simultaneity into each program,
2. creating programs which operate under control of the Series 200 Operating System.

Specifically, 1401, 1440, and 1460 programs are automatically translated into Series 200 programs which operate under control of the Mod 1 Operating System. Programs written for the 1410 can be translated to operate under control of the Mod 2 Operating System. Translated programs can be combined with new Series 200 programs, allowing the user complete flexibility in selecting programs to perform a particular job.

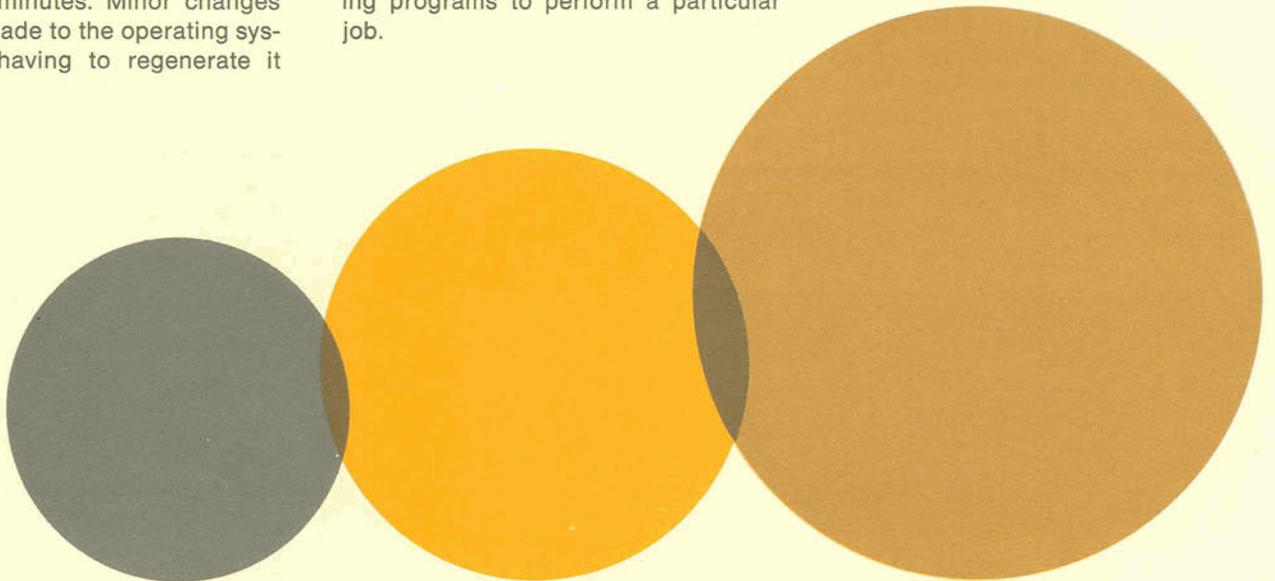
Is the Operating System Modular in Design?

A modular design can insure an economical solution to the operating requirements of many different installations. The Series 200 Operating System is divided into three models to insure that at all levels of installation size and complexity there is an operating system which affords the user a full complement of operating functions with minimal equipment overhead requirements.

For the small-to-medium-scale system, Mod 1 keeps overhead to a minimum by providing decentralization of such functions as input/output control. Under this approach, source-program requests for input/output operations are replaced by specialized control routines at program assembly or compilation time.

For the medium-to-large-scale user, throughput efficiency is maximized by centralizing all input/output control. The same input/output requests can be written in the source program without modification, but in this case they are specialized at program execution time.

For the user of Honeywell's super-system, the Model 8200, Mod 8 insures optimum utilization of the system's extensive multiprogramming/multiprocessing capabilities.



Series 200 Operating System In Review

Mod 1

Available in either a magnetic tape or a mass-storage version, Mod 1 provides a powerful computer management system for users of Model 120, 200, 1200, and 2200 computers. The tape-resident version operates on a computer having a core memory capacity of from 12K to 128K characters and from one to six tape drives. Flexible design of the operating system permits efficient use of random-access files, data communication, punched paper tape, and card devices in both independent and semi-centralized operations.

The mass-storage-resident version of Mod 1 operates on computers having a random-access device and as little as 8,192 characters of core memory. The system permits use of magnetic tape, punched card, and paper tape devices, as well as additional random-access units.

Mod 1 offers the user the following important advantages:

- Relieves the operator of detailed and burdensome execution supervision.
- Permits operation with only the needed functions and features.
- Provides execution of stacked jobs without operator intervention.
- Assures maximum use of the central processor and peripheral units through multiprogramming of data conversion or real-time applications and a major data processing job.
- Assures program compatibility as system grows.
- Allows tailoring and specialization of precoded, fully tested library routines.

Mod 2

Mod 2 provides the interface between the user and Honeywell's Model 1200, 2200, and 4200 computers. The operating system utilizes the following minimum equipment configuration:

1. A central processor equipped with 49,192 characters of memory and the Optional Instruction Feature.
2. Five magnetic tape units or three magnetic tape units and one random-access storage unit.
3. One card reader or an additional tape unit.
4. One printer.
5. One console.

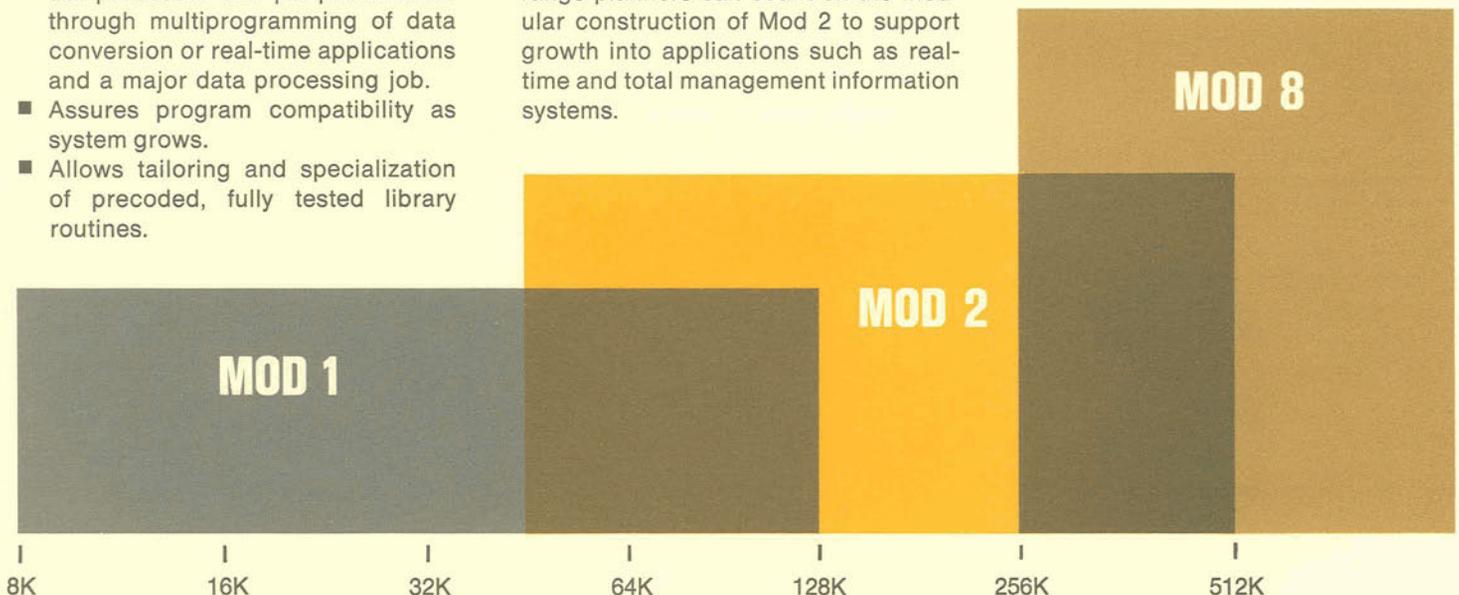
From the perspective of the data processing manager, the convenience and modularity of the Mod 2 Operating System is reflected and amplified in the over-all efficiency and reliability of the hardware/software complex. Standardized programming and operating procedures provide the most efficient path from initial formulation of a programming problem to final utilization of the solution. Total hardware utilization through multiprogramming and reduced idle time because of automatic job-to-job transition increase throughput under the Mod 2 Operating System. In addition, stacked-job processing provides comprehensive software service with minimal turnaround time. Long-range planners can count on the modular construction of Mod 2 to support growth into applications such as real-time and total management information systems.

Mod 8

The Mod 8 Operating System is a multiprogramming/multiprocessing software system which provides integrated operation control, program preparation, system support, and special application functions for the Model 8200 computer. Mod 8 can control the simultaneous execution of up to nine programs by harnessing the computer's built-in multiprogram control functions and its capability for simultaneous processing of both word- and character-oriented programs.

Both production and checkout programs may be executed simultaneously without danger of interference. A comprehensive system of "locks" and "keys" limits the access and storage activities of each program to its designated file and memory space.

Within the multiprogramming/multiprocessing capability of the Mod 8 Operating System, the user can incorporate "multitask" operations in his programs. This technique makes it possible to execute interdependent subprogram operations in parallel. The mechanics of multitask processing, which include recognizing and resolving all time and resource interdependencies, are automatically handled by the Mod 8 Operating System.



Three Generations of Operating Experience

Since the earliest days of commercial data processing, computer users and manufacturers alike have sought ways to improve the efficiency of computer operations. By reducing the need for human intervention, by making more productive use of available processing time, and by using all of the computer's resources to maximum advantage, operating systems have steadily progressed toward this goal. Honeywell has consistently played a leading role in developing operating systems to help users obtain the greatest payoff from their investment in data processing equipment. The experience gained from three generations of operating system development has enabled Honeywell to offer users at all levels a full complement of automatic operating services with truly minimal equipment requirements.

TODAY
Series 200
Operating System

1963
ADMIRAL
OPERATING
SYSTEM

One of the first operating systems to incorporate dynamic-stacked job scheduling.

1960
EXECUTIVE SYSTEM

First multiprogramming operating system — controls simultaneous execution of up to seven programs.

1957
D-1000 OPERATING SYSTEM

First of its kind to employ file updating techniques for program checkout operations.

Honeywell

ELECTRONIC DATA PROCESSING

SALES OFFICES AND DATA CENTERS
IN PRINCIPAL CITIES OF THE WORLD