File No. S360-20
Form C28-6534-1

OS

IBM Systems Reference Library

IBM System/360 Operating System

Introduction

This publication describes the general organization, function, and application of IBM System/360 Operating System.

The operating system is designed to extend the performance and application of Computing System/360 and to assist the manager, programmer, and operator of the system. The operating system consists of a comprehensive set of language translators and service programs operating under the supervisory control and coordination of an integrated set of control routines. It is designed for use with Models 30, 40, 50, 65, and 75 of Computing System/360. The operating system is located in direct-access storage, such as disk or drum, and operates on a computing system having 32,768 bytes or more of main storage.

PREFACE

The purpose of this publication is to serve as a general introduction to IBM System/360 Operating System. Since many combinations of programming and computing facilities are possible with IBM System/360, no attempt is made in this summary to relate the operating system facilities to specific machine requirements. More comprehensive information may be obtained from IBM representatives or by reference to the publications described in the last section of this publication. Abstracts of other System/360 publications are contained in the IBM System/360: Bibliography, Form A22-6822.

MAJOR REVISION (November, 1966)
This publication is a major revision of the previous editicn, Form C28-6534-0, which is now obsolete. The section "Publications" has been rewritten to reflect the more complete set of publications prepared for IBM System/360 Operating System. In addition, minor changes have been made throughout.

This publication was prepared for production using an IBM computer to update the text and to control the page and line format. Page impressions for photo-offset printing were obtained from an IBM 1403 Printer using a special print chain.

Copies of this and other IBM publications can be obtained through IBM Branch Offices.

A form for readers' comments appears at the back of this publication. It may be mailed directly to IBM. Address any additional comments concerning this publication to the IBM Corporation, Programming Systems Publications, Department D58, PO Box 390, Poughkeepsie, N. Y. 12602

FIGURES

IBM System/360 Operating System consists of a comprehensive set of language translators and service programs operating under the supervisory control and coordination of an integrated set of control routines. The operating system is designed for use with Models 30, 40, 50, 65, and 75 of Computing System/360. The operating and computing systems are an integrated whole. One is an extension of the other, and together they form a single system that is operated, programmed, and managed as an 'entity (Figure 1).

Capabilities of the operating system can be tailored to the needs at individual installations. In many cases, the customer can select from several of each of the major components of the operating system having different functional capabilities. Also, many operating system components have optional features that the customer can select according to his needs. Thus, not all of the facilities described in this publication will be selected at all installations.

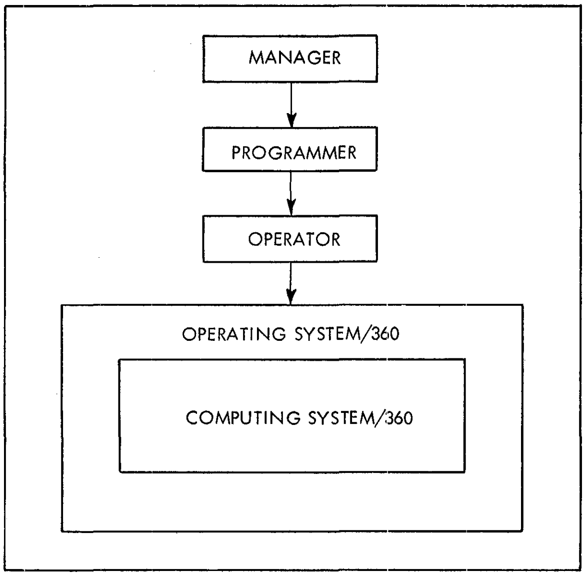

Figure 1. System/360

MAJOR OBJECTIVES

IBM System/360 Operating System is designed:

- To increase the throughput of Computing System/360, that is, the total volume of work performed by the system over a period of time.
- To decrease the response (or turn-around) time of the system, that is, the interval between the time a unit of work is submitted for processing, and the time that a response (or result) is received from the system.
- To assist the manager of an installation.
- To assist in programming an application of the system.
- To assist the operator of the system.
- To enable a flexible and orderly growth of the system through a broad range of applications.

INCREASED THROUGHPUT

One way in which the throughput of System/360 is increased by the operating system is through its ability to process a continuous stream of jobs without interruption. This is accomplished by providing automatic transition from one job to another and by enabling the operator of the system to mount removable input/output volumes (e.g., tape reels and disk packs) for one job while previously scheduled jobs are being processed.

Increased throughput can also be achieved in System/360 by more efficient use of the physical and programming resources of the system. For example, an optional system of control enables programs, storage space, input/output facilities, and control of the central processing unit to be allocated and concurrently shared among several data processing tasks. The operating system helps to ensure that as much of the total system as possible is kept busy performing productive work as much of the time as possible. This is accomplished by efficiently allocating the available resources of the system to more than one task, and switching control from one task to another as a delay is encountered while awaiting an event (such as the completion of an input/output operation or the end of a timing interval).

DECREASED RESPONSE TIME

The operating system decreases response time in System/360 primarily by limiting and controlling the degree of human participation in the mechanics of data processing, and by providing more direct and more automatic lines of communication and control between the work source and the system. When jobs are received locally, they can be automatically processed from beginning to end on a single system. Input/output conversion operations, such as converting from punched cards to direct-access storage, can be performed efficiently on a single system concurrently with the processing of the jobs. Therefore, the delays and human activity involved in performing such operations on an off-line peripheral system are eliminated.

Even more rapid response is possible for some applications. The full data processing and programming facilities can be extended, by way of telecommunication lines, to remote locations, thereby enabling the system to become a direct and integral part of the activity it supports.

ASSISTANCE TO THE MANAGER

The operating system assists the manager of an installation by providing facilities for the following:

- Dividing the programming effort to shorten the time between the definition and solution of a problem.
- Scheduling work in accordance with priorities assigned at the installation.
- Keeping an inventory of, and controlling the use of, data and programs that are stored within the system.
- Maintaining system programs efficiently.

ASSISTANCE TO THE PROGRAMMER

The operating system assists the programmer in programming a problem solution or an application by a variety of means, such as:

- Providing programming aids that enable a problem solution or an application to be expressed in languages and terms that can be readily understood.
- Providing easy and immediate access to often used precoded routines and programs.
- Providing facilities for easily storing, retrieving, testing, and modifying programs and data.
- Providing a system of control that relieves the programmer of concern about the current status of the computing system.

ASSISTANCE TO THE OPERATOR

The operating system assists the operator by:

- Providing specific instructions.
- Providing facilities to communicate with and direct the system.
- Performing automatically, and therefore more quickly and reliably, functions such as logging errors and keeping track of programs and data.

FLEXIBLE AND ORDERLY GROWTH

Flexible and orderly growth of the operating system through a broad range of applications and computing system configurations is achieved by:

- Providing a variety of programming facilities.
- Providing a modular design having a common system of intercommunication and control that enables the system's facilities to be combined in a variety of ways.
- Providing an open-ended design that enables customers' programs and data to be easily incorporated into the system.
- Enabling programs to refer to data and other programs without regard to the specific input/output configuration of the system.
- Ensuring programming compatibility as the system grows.

Although an operating system provides a number of distinct advantages in terms of performance, utility, and application, it also requires space in main and auxiliary storage, as well as processing time to perform its functions. Many of the services furnished by an operating system are basic (e.g., program loading and input/output control). These would require storage space and time to be performed whether or not an operating system were employed to perform them. Other services provided by an operating system result in better performance, wider application, or a reduction in manpower and training costs. The benefits of an operating system depend to a large extent on the specific requirements of an installation and how closely the services provided by the system meet those requirements. If a facility provided by an operating system is not used at a particular installation, it should not take up storage space. Therefore, System/360 Operating System is based on a modular design that enables many facilities to be selected on the basis of whether or not they are required at a particular installation.

MODULAR DESIGN AND CONSTRUCTION

The operating system is a composite of many program modules, which, like the physical facilities of Computing System/360, can be united in a variety of combinations because they adhere to commonly established conventions for intercommunication and control. The number, types, and versions of control routines and programming facilities employed in the operating system vary depending upon the exact requirements at a particular installation. Those modules that perform optional functions need not be included in a given system. Each operating system consists of a selection of control routines and programming facilities that are closely integrated with a selection of processing, storage, and input/output facilities to form a balanced system for a particular range of applications.

Means are provided for generating an operating system for a particular installation from a definition of the computing and operating system facilities selected. The operating system is generated using standard language and editing facilities of the system.

GROWTH IN PERFORMANCE AND APPLICATION

As data processing requirements at an installation increase, the operating system, as well as the computing system, can easily be expanded in both performance and application. The ability to grow is inherent in the design of both systems. It enables a smooth evolutionary expansion in application and performance to be achieved without destroying compatibility with existing programs and applications. A job that is processed on one System/360 can be processed on another, provided, of course, the minimum operating and computing system facilities required by the job are present.

SHORT-TERM REQUIREMENTS

Because data processing requirements may vary from day to day or hour to hour, the operating system is designed to adapt to short-term as well as long-term changes in requirements. The programs that constitute an operating system for an installation are kept in libraries located in direct-access storage. However, all library programs need not be immediately accessible when the system is operating. Library programs that are infrequently used can be located on a removable storage volume, such as a disk pack.

The programs provided with the operating system are, in general, designed for short-term adaptability to requirements in that many key operating characteristics can be modified to satisfy the requirements of a particular job. Normally, however, in the absence of explicit specifications by the programmer, the programs operate in a mode designed to satisfy the requirements of the majority of jobs. Therefore, for most jobs, very few options need be specified by the programmer, although many are available.

PROBLEM PROGRAM ADDITIONS

Once an operating system is generated, it can be easily expanded to include a user's programs and data. Such programs may be language translators or service programs, similar to those supplied by IBM, or they may be a user's production programs. Programs supplied by IBM and a user's own programs are, in fact, equivalent in that they can be stored into and retrieved from libraries in the same way. They are also written using the same conventions for communicating with one another and with other parts of the operating system.

A user's program and data may be incorporated in the operating system for the duration of a single job or it may be stored in a library and remain a part of the system for an extended period of time for use in different jobs.

USERS' PROGRAM LIBRARIES

The operating system eliminates the need for maintaining separate libraries of users' programs in two forms -- with and without final main storage addresses assigned to the instructions. Usually, users' programs are stored in a form ready for execution except that main storage addresses have not been assigned. In this form, one program can be combined with one or more other programs to form a single larger program. In other systems, a second library of users' programs in which main storage addresses have been assigned is also required. In this form, programs can be loaded directly into main storage whenever their execution is desired. In the System/360 Operating System, final main storage addresses are assigned as part of the loading process. Thus, there is no need for keeping a library of users' programs in which final main storage addresses have been assigned.

NATIONAL AND INTERNATIONAL USE

IBM System/360 Operating System is designed for use throughout the world. It can handle Sterling and other currency conventions and can be easily modified to use national character sets and to communicate with the operator and programmer in languages other than English. These features not only enable the system to be easily tailored for use in a particular country, but also enable it to be adapted to data processing activities that are worldwide in scope.

TECHNOLOGICAL ADVANCEMENT

Both the operating system and the computing system are designed to take advantage of anticipated future developments and improvements in data processing technology. Conventions for intercommunication and control that are established in the basic design of the operating system will enable it to adapt to new techniques and equipment and thereby steadily grow in performance and application.

The IBM System/360 Operating System consists of a control program and a number of processing programs (Figure 2). In general, the control program governs the order in which the processing programs are executed and provides services that are required in common by the processing programs during their execution. The processing programs consist of language translators and service programs that are provided by IBM to assist the user of the system, as well as problem programs that are written by the user and incorporated as part of the system. Both IBM and user programs have the same functional relationship to the control program.

Information on the general use and functions of the facilities of the operating system is provided in the publication IBM System/360 Operating System: Concepts and Facilities, Form C28-6535.

PROCESSING PROGRAMS

A wide selection of IBM-supplied processing programs is available for inclusion in the operating system. These may be supplemented in the future by others supplied by users of System/360 or by IBM. The processing programs are designed to reduce the time, training, expense, and manpower required to prepare and execute efficient problem programs. The programmer may use them singly or in combination to process a particular job. Some of the processing programs are language translators. Others are generalized service programs, such as a sort/merge program. Most are available in different versions having different characteristics.

LANGUAGE TRANSLATORS

The language translators assist a programmer by enabling him to define a problem solution or an application in a language form that can be readily learned and understood. They relieve the programmer from the detailed work involved in programming problem solutions and thereby reduce the training and time required to produce efficient programs. Language translators are provided in the operating system for defining a problem solution or application:

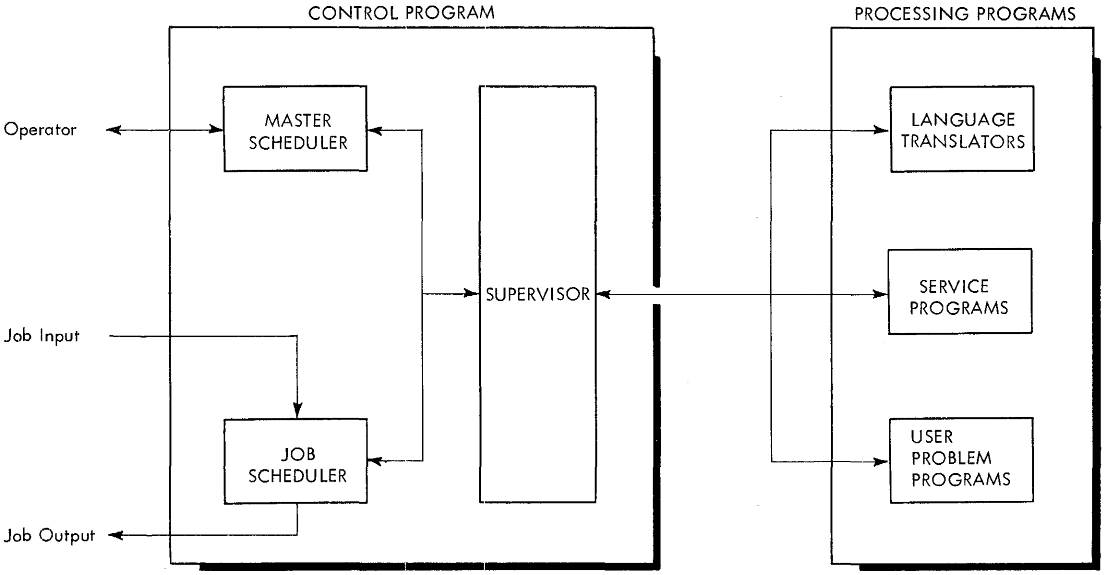

Figure 2. General Organization of Operating System/360

- In a form of mathematical notation (FORTRAN and ALGOL).

- In a concise form of the English language (COBOL).

- In a new programming language (PL/I) having features of both FORTRAN and COBOL.

- In a flexible, easy-to-use symbolic language (Assembler Language).

- In a tabular form (Report Program Generator Language).

Program testing facilities are provided in some of the language translators. These facilities enable a program or part of a program to be dynamically and selectively tested, in accordance with simple and concise specifications expressed in the source language.

FORTRAN Compilers

Compilers can be included in the operating system for use in compiling object programs from source programs written in the FORTRAN language. The FORTRAN language is a widely accepted and used language that was developed and refined over a period of years through the combined efforts of IBM, its customers, and the American Standards Association. It closely resembles the language of mathematics and enables engineers and scientists to define problem solutions in a familiar, easy-to-use notation.

The FORTRAN language is described in the publications IBM System/360: FORTRAN IV Language, Form C28-6515 and IBM System/360 Operating System: Basic FORTRAN IV Language, Form C28-6629.

COBOL Compilers

Compilers can be included in the operating system for use in compiling object programs from source programs written in COBOL. COBOL is based on a well defined, restricted form of English that provides a convenient method of producing programs to solve commercial data processing problems.

COBOL is a widely used language that was developed as a cooperative effort by a number of computer manufacturers and users under the direction of the Conference on Data System Languages (CODASYL) sponsored by the Department of Defense. IBM Operating System/360 COBOL can provide the following major facilities:

- Report Writer.
- Sort.
- Source Program Library.
- Mass Storage.
- Sterling Conversion.

The COBOL language is described in the publication IBM System/360 Operating System: COBOL Language, Form C28-6516.

ALGOL Compiler

A compiler can be included in the operating system that enables scientific and technical problems to be stated in the international algorithmic language, ALGOL. The algorithmic language permits computational processes to be described by arithmetic expressions containing numbers, variables, and functions.

The ALGOL language and procedures for performing input/output operations are described in the publication IBM System/360 Operating System: ALGOL Language, Form C28-6615.

PL/I Compiler

A compiler can be included in the operating system for use in compiling object programs from source programs written in PL/I (Programming Language/I). This language incorporates some of the best features of other higher-level languages as well as new features, for instance, string manipulation, data structures, dynamic storage allocation, and extensive editing capabilities.

PL/I takes full advantage of recent developments in computer technology. It provides the programmer with a problem-oriented language for efficiently programming either scientific or commercial problems, as well as problems that can best be solved with a combination of scientific and commercial computing techniques. It is particularly useful for the increasing number of applications having both commercial and scientific characteristics, such as information retrieval or command and control applications. The modern features of PL/I enable it to be used for many programming applications for which other compiler languages either cannot be used, or can be used only with considerable difficulty.

PL/I is described in the publication IBM System/360 Operating System: PL/I Language Specifications, Form C28-6571.

Assemblers

Assemblers can be included in the operating system for use in assembling object programs from source programs written in a flexible and easy-to-use symbolic language. The assembler language is a versatile machine-oriented language that can be used for any type of application.

A macro-language is provided as an extension of the assembler language. The macro-language provides the programmer with a convenient way of defining a sequence of assembler language statements that can be used many times in a program. The macro-definition is written only once, and only a single statement, a macro-instruction statement, is written each time a programmer wants the assembler to generate the defined sequence of assembler language statements. A macro-definition may also be incorporated as part of the operating system and be used by many different programs.

A comprehensive set of system macro-definitions is provided for communicating service requests to the supervisor, for communicating with input/output routines, and for testing programs.

The assembler language is described in the publication IBM System/360 Operating System: Assembler Language, Form C28-6514. The system macro-instructions are described in the publication IBM System/360 Operating System: Control Program Services, Form C25-6541.

Report Program Generator

The report program generator provides the programmer with an efficient, easy-to-use facility for generating object programs that are used to produce reports from sets of data. The reports may range from a simple listing of a card deck to a precisely arranged and edited tabulation of calculated data from several input sources. Several reports can be created concurrently from a single set of data.

The report program generator is described in the publication IBM System/360 Operating System: Report Program Generator Specifications, Form C24-3337.

SERVICE PROGRAMS

The service programs assist a programmer by providing routines for performing frequently used functions; and also by providing routines for editing, linking, and otherwise manipulating programs and data. The service programs consist of linkage editors, a sort/merge program, and a set of utility programs.

Linkage Editor

A linkage editor is provided for use in combining program segments that were individually compiled or assembled. It forms a single program that is ready to be loaded into main storage and executed. The linkage editor enables changes to be made in a program without recompiling (or reassembling) the complete program; only those sections that are changed need be recompiled. The linkage editor also permits a program that is too large for the space available in main storage to be divided, so that executed segments of the program can be overlaid by segments yet to be executed.

The linkage editor and its use are described in the publication IBM System/360 Operating System: Linkage Editor, Form C28-6538.

Sort/Merge Program

The sort/merge program is a generalized program that can be used to sort or merge fixed- or variable-length records in ascending or descending order. The sorting and merging can be performed using magnetic-tape and direct-access storage devices for input, output, and intermediate storage. The program takes full advantage of the input/output resources that are allocated to it by the control program. The sort/merge program can be used independently of other programs, or it may be used directly by programs compiled by COBOL compilers.

The sort/merge program and its use are described in the publication IBM System/360 Operating System: Sort/Merge, Form C28-6543.

Utility Programs

A comprehensive set of utility programs is provided in the operating system for performing functions such as:

- Transferring (and in the process, changing the format and editing) data from one storage medium or input/output device to another.

- Editing, rearranging, and updating programs and data in the system library.

- Changing the indexing structure of the system library catalog.

- Printing an inventory of the data and programs that are cataloged in the system library.

The utility programs and their use are described in the publication IBM System/360 Operating System: Utilities, Form C28-6586.

GRAPHIC PROGRAMMING SERVICES

Programming services are provided with the operating system to simplify the processing, display, and retrieval of information in graphic form. These services can be used in a variety of advanced applications in the fields of science, engineering, and business to perform such operations as:

- Displaying information and checking its accuracy.

- Retrieving and displaying graphic information previously entered into the system, modifying it at a display screen, and re-entering it.

Graphic programming services simplify preparation and execution of programs that employ either of the following devices attached to System/360:

- IBM 2250 Display Unit
- IBM 2260 Display Station

Graphic programming services include an extensive set of macro-instructions, problem-oriented routines, data-manipulation aids, input/output interruption control and analysis routines, and error recovery and diagnostic routines.

Additional information on graphic programming services can be obtained from the publications IBM System/360 Operating System: Express Graphic Programming Services for IBM 2260 Display Station (Local Attachment), Form C27-6925, IBM System/360 Operating System: Express Graphic Programming Services for IBM 2250 Display Unit, Model 1, Form C27-6921, IBM System/360 Operating System: Basic Programming Services for IBM 2250 Display Unit, Form C27-6909, and IBM System/360 Operating System: Programming Services for IBM 2260 Display Station (Local Attachment), Form C27-6912.

CONTROL PROGRAM

The control program is designed to control the processing of a continuous series of jobs, the work flow within the system, and input/output operations.

The job is the basic independent unit of work performed by the operating system. Each job is independent of any other job and consists of one or more directly or indirectly related steps. Each job step results in the execution of a processing program, such as a language compiler, or a user's object program, such as a payroll program.

Some job steps are directly related to one another in that the output of one is passed on as the input to another. An example would be three successive job steps, in which the first results in the execution of a user's problem program; the second sorts the output of the first; and the third produces a report from the output of the second.

Other job steps are only indirectly related. An example of this would be two job steps, one of which compiles a portion of a source program written in one language, followed by a second that compiles a portion of a program written in another language.

Key parts of the control program are loaded into main storage and remain there indefinitely to ensure continuous coordinated operation of the system. Other parts of the system are brought into main storage from auxiliary storage as they are required to perform specific functions.

The control program includes three major parts: the job scheduler, the master scheduler, and the supervisor. In addition, a group of routines is provided which are brought into main storage only as they are required. The functions performed by these routines, together with some of the functions performed by the three major parts of the control program mentioned above, are collectively called data management. Data management is described following the descriptions of the three major parts of the control program.

JOB SCHEDULER

The primary functions of the job scheduler are:

- Reading job definitions from one or more input/output devices assigned as job input sources.
- Allocating input/output devices to each job step, when necessary.
- Initiating the execution of the processing program specified for each job step.
- Processing selected output produced during each job.
- Providing records of work processed.

There are several job schedulers available, ranging from a scheduler that processes jobs sequentially to schedulers that enable more than one job and more than one job step to be processed concurrently and thereby share the resources of the system. Some job schedulers can read jobs concurrently from several input devices, such as card readers, and can record output on several output devices, such as printers and card punch units. The reading and recording can be performed either independently of, or concurrently with, the actual processing of the jobs. A priority job scheduler can also schedule the order in which jobs are processed, based on the priorities assigned to the jobs and the availability of the physical resources of the system.

Information on the control statements that are used by the programmer to define a job may be obtained from the publication IBM System/360 Operating System: Job Control Language, Form C28-6539. The publication IBM System/360 Operating System: Job Control Language Charts, Form C28-6632, provides assistance in coding the job control statements.

MASTER SCHEDULER

The master scheduler serves as a two-way communication link between the operator and the system. By issuing commands to the master scheduler, the operator can alert the system to a change in the status of input/output units, alter the operation of the system, and request information on the status of the system. The master scheduler is also used by the operator to alert the job scheduler to the source of jobs and to initiate the reading or processing of jobs.

Master schedulers are available that differ primarily in the number and types of operator commands that can be executed. Additional information on the master scheduler and the operator commands that it accepts may be obtained from the publication IBM System/360 Operating System: Operator's Guide, Form C28-6540.

SUPERVISOR

The supervisor is the control center of the operating system. Its primary function is to perform a variety of services for other parts of the system including problem programs. It coordinates and controls the performance of these services to ensure efficient and coordinated use of the physical and programming facilities, or resources, of the system. The supervisor prevents programs and routines that are run on the system from interfering with one another and with the operation of the control program. This is accomplished, in part, through its use of privileged instructions, such as storage-protected and input/output instructions, which can be executed only by the control program.

To perform its functions, the supervisor receives control of the central processing unit by way of an interruption. The interruption may result from a specific request for services from another part of the operating system or from a problem program, or it may be an automatic interruption, such as an interruption that occurs at the completion of an input/output operation. The automatic interruptions, in general, enable the supervisor to coordinate its control over the physical and programming resources of the system.

A service performed by the supervisor may be specifically requested by a program, such as a request for storage space, or it may be a service that is automatically provided when a contingency occurs, such as attempting to recover from an error condition. Among the services that may be provided by the supervisor are the following:

- Allocating main storage space required by programs during their execution.
- Sharing areas of main storage among routines that need not be in main storage at the same time.
- Loading programs into main storage.

- Controlling the concurrent execution of programs and routines.
- Scheduling and controlling input/output operations.
- Providing the time of day and other timing services, such as keeping track of the time at which a particular operation is to be performed.
- Providing standard procedures that assist in diagnosing exceptional conditions, such as underflow in floating-point arithmetic operations.
- Keeping a running log of input/output unit errors for Customer Engineer diagnostic use.

Parts of the supervisor are available in different versions having different characteristics. Some parts are required; others may not be required depending on the application of the particular system.

MAJOR FUNCTIONS

The control program as a whole is designed to perform the following three major functions:

- Data Management, which enables all types of data and programs that are processed by System/360 to be systematically and efficiently organized, identified, stored, cataloged, and retrieved.
- Job Management, which enables a continuous series of jobs to be performed by System/360 with little or no operator intervention.
- Task Management, which enables several data processing tasks to be performed concurrently and thereby increases the total throughput of System/360.

Each of these functions is described in the following paragraphs.

DATA MANAGEMENT

In a modern data processing installation, a major problem is that of managing the mass of data and programs processed within the installation. To help solve this problem, the operating system contains data management facilities designed to:

- Permit a programmer to store, modify, and refer to programs and data using the compact and economical direct-access storage facilities of the system (such as disk storage) without regard to the specific input/output configuration of the system.
- Permit the free interchange and use of programs and data among programmers.
- Provide a systematic method for identifying and locating programs and data, and systematic methods for referring to the data after it is located.

SYSTEM LIBRARY

The complete operating system (including problem programs as well as data processed by the problem programs) is stored, as an organized library, in as many auxiliary storage volumes as required. The library has a built-in catalog that may be used to identify and locate any type of data. As the operating system grows, and more and more programs and data are stored in the library, the catalog can be extended to reflect this growth. The library catalog enables each programmer to store his own private data as part of the library and later retrieve it by a unique alphameric name.

The library consists of collections of data called data sets. A data set is made up of one or more blocked or unblocked records that are logically related.

In general, no distinction is made within the library as to the type of data. Data is cataloged in the same way whether it is a set of job control statements, a source program, an object program, or a set of data records to be processed by an object program. The only distinction among the different types of data is in the nature of the data itself and the way in which it is used; for example, whether it is a source program processed by a language compiler, an object program segment processed by the linkage editor, or a set of data records processed by a user's object program.

Means are provided for easily storing and cataloging within the library any of the several types of data. Using the library and direct-access storage facilities of the system, a programmer can store, test, modify, recompile, link, and execute his programs within the confines of the operating system. Therefore, it is not necessary to maintain large files of punched cards.

DATA SET CATALOGING SYSTEM

The cataloging system of the library may be compared to the Dewey decimal library classification system. Alphameric names of up to eight characters, rather than numbers only, are used to identify a set of data. In this system, a data set name may be referred to as follows:

DESIGN.ELECTRO.ROBERTS

where ROBERTS is the basic name of a data set that is classified under the name ELECTRO that, in turn, is classified under the name DESIGN.

Means are provided for a user to construct a library catalog consisting of several levels of indexes. The catalog structure for a particular operating system library may be represented, as shown in Figure 3, in the form of an organization chart. The data sets within the library can be classified in many ways; for example, to reflect the organizational structure of an engineering department that uses the system.

It is possible to construct a catalog on a removable direct-access storage volume, such as a disk pack. For example, the first level index named DESIGN in Figure 3, as well as the indexes under it, might be stored on a removable disk pack which is assigned for exclusive use by the design department.

DATA SET GENERATIONS

In addition to cataloging single data sets within the library, several successive generations (updates) of a data set can be cataloged in the library. This method may be used to catalog a data set, such as year-to-date earnings, that is updated weekly by a payroll program. Each generation of the data set may have the same name and be referred to relative to the current generation of the data set. With this method of cataloging, the system can automatically keep track of the generations that have been produced.

DATA SET CONTROL AND MODIFICATION

A comprehensive set of utility programs is provided in the operating system to help the manager of an installation control both the classification of data within the library and the use of auxiliary storage. Some of these programs are used for creating, renaming, or deleting indexes or data sets; for reorganizing and rearranging the data set library; and for performing periodic surveys to determine the organization and contents of the library. Other utility programs are provided for the use of individual programmers in updating their data sets.

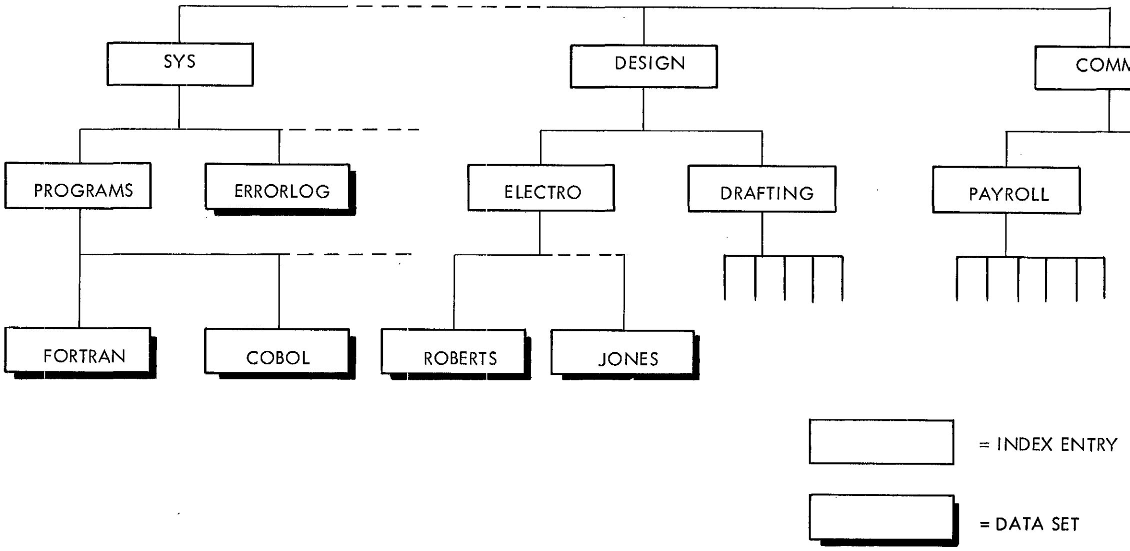

Figure 3. Simplified Diagram of a Catalog Organization

DATA SET ACCESS

Input/output routines are provided in the operating system to relieve the programmer from the burden of writing complex routines for each program that processes data sets. These routines automatically perform functions such as overlapping processing with input/output operations, and preparing and checking labels. Input/output routines are available for:

- Processing logical or physical records stored on input/output devices in sequential order.
- Processing logical or physical records stored on input/output devices in nonsequential order.
- Processing messages received from remote locations at unpredictable intervals of time.

A description of each data set is normally stored in auxiliary storage along with the data set itself. The description provides the input/output routines with a variety of information about the data set, such as its organization, physical location, and size. The data set need not be redescribed each time it is referred to by a program. However, means are provided for changing the description for a particular program, or for a particular execution of a program, without reassembling or recompiling the program. For example, the size of the input or output buffer area to be used in processing a data set may be changed for a particular execution of a program.

DEVICE INDEPEPNDENCE

The operating system has been designed so that a program processing a data set need not be changed because of the input/output device type used for the data set. For example, the operating system may, depending on the availability of input/output devices, process a data set representing a report by printing it directly on an on-line printer; recording it on magnetic tape for later off-line printing; or recording it in disk storage for later on-line printing. Therefore, a programmer need not be concerned with the availability of a particular device or device type, and a program can be executed on systems with different devices or device combinations.

Additional information on data management may be obtained from the publication IBM System/360 Operating System: Data Management, Form C28-6537.

JOB MANAGEMENT

An important feature of an operating system is its ability to process a continuous series of jobs with little or no operator intervention. By reducing the degree of human participation in the mechanics of data processing, the operating system ensures that jobs are processed faster and are less subject to human error. As a result, turn-around time (the interval between the time a programmer submits a job for processing and the time he receives results) can be significantly reduced, and the throughput of the system can be significantly increased.

In planning a job, the programmer may employ any logical combination of the many programs that may be included in the operating system library. In effect, he controls and directs the processing of his job from his desk by using the proper job control statements. If the job control statements are to be used repeatedly with little or no change, he can store them within the system library and thereafter refer to them by using a single job control statement.

A job may take many forms. It may be a job that results in compiling and cataloging a source program or it may be a complex commercial job that has several steps involving a number of different processing programs and the passing of data sets from one step to another. A job may be of relatively short duration or it may last for hours, such as a job that processes real-time transactions or inquiries received from remote terminals.

Before a particular job is processed, it may be combined with other jobs to form a single input batch. To ensure more efficient processing, the batch may be recorded on magnetic-tape or direct-access storage preparatory to the actual processing. This may be done either off-line on a Computing System/360 functioning as an auxiliary to another System/360, or it may be done on-line, concurrently with job processing. The jobs may have been received originally from local sources, in the form of punched cards, or they may have been received from a remote terminal.

As the jobs are processed, the output produced by the operating system can be stored in a common output batch on magnetic-tape or direct-access storage. The output batch may contain object programs compiled or assembled by the operating system, source program listings, program test listings, messages for the programmer from the operating system, and output from problem programs. When an output batch is completed, it is processed to produce printed listings or object program card decks that are distributed to the programmers who initiated the jobs. An output batch may be processed either on-line, concurrently with the job processing, or off-line on an auxiliary system.

The exact way in which input and output batches are formed and processed at a particular installation depends on the job scheduler included in the operating system.

A programmer may place an input data set for an object program within his job and define it as being part of the common input batch. Similarly, output data sets produced by the program can be defined as being part of a common job output batch. As a result of defining data sets in this way, no setup delays are incurred within the job, and input/output units that would otherwise be required are available for other purposes.

The operator of the system, for the most part, performs relatively routine functions, such as loading and unloading tape reels. If the operating system cannot complete a job or job step because of a programming error, it automatically skips to the next job or job step without intervention by the operator. However, the operator can, at any time, communicate with the operating system or direct it to perform certain functions. For example, he may indicate a change in the status of an input/output device or in the priority of one of the jobs waiting to be processed.

TASK MANAGEMENT

To perform a data processing task, a number of basic computing system facilities or resources are required. Input/output or auxiliary storage devices are required for entering or storing input data and for recording or storing output data. Time is required on input/output channels for transmitting to and from main storage and for initiating and controlling input/output operations. Main storage space is required for storing a series of instructions and the data processed by the instructions. Finally, central processing unit time is required for executing the instructions that process the data.

In most current computer installations, data processing tasks are performed sequentially; that is, a new task is not begun until the current task is completed. The average data processing task performed by a computing system requires at any given moment only a fraction of the total available resources of the system. Therefore, many parts of the system are often idle for significant periods of time. For example, many data processing tasks, such as a data conversion task, require only a fraction of the storage space and input/output devices available in the system and intermittent use of the central processing unit.

To increase the throughput of System/360, optional task management facilities are available for inclusion in the control program. These facilities permit multiprogramming; that is, they enable data processing tasks to be performed concurrently and to share the basic resources of the computing system. Thus, if a resource, such as main storage space, is momentarily not required by one task, it can be allocated to another and thereby improve the overall efficiency of the system. In addition to the facilities for performing tasks in a normal sequential manner, optional facilities are available for concurrently performing several tasks within a single job step or for concurrently performing several tasks within steps of different jobs.

With the optional facilities, since one data processing task can begin before another is completed, it is possible to concurrently perform several tasks using a single routine that satisfies certain restrictions. Such a routine, if it is repeatedly and concurrently required by many different tasks, need not be duplicated in main storage. This not only saves main storage space but reduces the time spent in loading routines into main storage.

In defining a data processing task, a programmer need not be directly concerned with the allocation of the resources required to perform the task. The task management facilities of the control program can perform functions such as those listed below. The last three functions are required only in systems that allow concurrent data processing tasks.

- Allocating space in main storage.

- Loading programs and routines into main storage.

- Scheduling the use of programs and routines in main storage.

- Switching control of the central processing unit from one task to another.

- Preventing one task from interfering with another.

- Controlling the execution of the tasks in accordance with a defined order of priority.

TELEPROCESSING

The operating system can be used to process messages received from remote locations by way of communication lines and telecommunication equipment. A message received from a remote location may be a request to the system for a particular service and may or may not be accompanied by data. The requested service may be simply the routing of a message to another remote location or it may be the processing of a job or transaction similar to jobs and transactions that are received locally.

The operating system may be used for a wide variety of teleprocessing applications, including the following:

- Message Switching, in which messages received from one remote terminal are sent to one or more other terminals.
- Job Processing, in which jobs received from remote locations are routed and assembled for processing by the job scheduler.
- Inquiry or Transaction Processing, in which inquiries or transactions received from a large number of widely separated locations are routed for processing by user-supplied message-processing programs.
- Data Collection, in which data received from a number of terminals is collected and stored for later processing.

The operating system may perform either job processing or teleprocessing. However, if optional task management facilities are incorporated into the system, both types of processing can be performed concurrently on the same system, instead of separate special-purpose systems.

The design of the operating system allows for a gradual transition from one type of processing to another. For example, a system may be applied initially to processing jobs, data, and transactions that are entered locally. Then, by later adding communication facilities, the services of the system can be gradually extended to the points of origin of the jobs, data, or transactions. A system can also be used to perform one type of processing during part of the day and then change over to perform another type of processing.

By extending the services of the system directly to the user via communication lines, the response time of the system can be reduced from hours to minutes or seconds. Consequently, the system can directly participate in and control various commercial and scientific activities as they are being carried on. For example, the system may be used to serve, from a central location, a geographically dispersed banking activity. In such a system, master files containing account records for thousands of depositors are stored in direct-access storage. By entering pertinent data into the system, tellers at widely separated locations can check balances, update passbook records, and handle similar transactions, all within a few seconds.

The system can also be applied to control similar activities in other industries such as insurance and sales. These activities may be carried out within a single building, be nationwide, or even worldwide in scope. In the fields of education, engineering, and research, the system may be used to extend the use of advanced programming and computing facilities to the desk of the student or engineer.

Teleprocessing differs from more conventional types of data processing mainly in the way in which information enters and leaves the system. Messages from remote locations enter the system in random order at unpredictable intervals and often demand a response from the system within a specified period of time. To supervise the sending and receiving of messages to and from remote terminals, a message control system may be constructed by the user. A set of message editing and control routines is provided for this purpose.

Additional information on teleprocessing applications can be obtained from the publication IBM System/360 Operating System: Telecommunications, Form C28-6553.

A set of IBM System/360 Operating System publications has been designed to provide the reader with the information he requires to do his particular job (e.g., operating the system or writing problem programs). The family of publications is, therefore, classed according to primary audience. In many cases, however, it is anticipated that readers having a particular responsibility will read publications intended for a different primary audience. For example, systems programmers will probably read many of the publications directed primarily toward applications programmers.

CONSOLE OPERATOR PUBLICATION

The following publication is provided for the console operator.

OPERATOR'S GUIDE

Information about operating IBM System/360 using the operating system is contained in IBM System/360 Operating System: Operator's Guide, Form C28-6540. This publication contains operating system initialization procedures, input/output device assignment procedures, job control information, descriptions of operator commands, and their use, and on-line messages with their explanations.

APPLICATION PROGRAMMER PUBLICATIONS

The following publications are directed to the applications programmer. They provide him with information about the languages he may choose and the various operating system facilities at his disposal.

CONCEPTS AND FACILITIES

The general concepts involved in the full use of the operating system are described in IBM System/360 Operating System: Concepts and Facilities, Form C28-6535. It describes the basic concepts of IBM System/360 Operating System and guides the programmer in the use of its various facilities.

DATA MANAGEMENT

A description of the data management facilities of the operating system is contained in IBM System/360 Operating System: Data Management, Form C28-6537. This publication contains detailed information about the cataloging, space allocation, and data access features of the operating system. Information is also included on record and label formats and on data organizations.

LANGUAGE PUBLICATIONS

The publications that provide information about the languages are:

IBM System/360 Operating System: ALGOL Language, Form C28-6615

IBM System/360 Operating System: Assembler Language, Form C28-6514

IBM System/360 Operating System: COBOL Language, Form C28-6516

IBM System/360: FORTRAN IV Language, Form C28-6515

IBM System/360: Basic FORTRAN IV Language, Form C28-6629

IBM System/360 Operating System: PL/I Language Specifications, Form C28-6571

IBM System/360 Operating System: Report Program Generator Specifications, Form C24-3337

LANGUAGE PROGRAMMER GUIDES

The publications that provide guidance to operating system users writing programs in the available languages are:

IBM System/360 Operating System: Assembler (E) Programmer's Guide, Form C28-6595

IBM System/360 Operating System: Assembler (F) Programmer's Guide, Form C26-3756

IBM System/360 Operating System: COBOL (E) Programmer's Guide, Form C28-6380

IBM System/360 Operating System: FORTRAN IV (E) Programmer's Guide, Form C28-6603

IBM System/360 Operating System: PL/I (F) Programmer's Guide, Form C28-6594

LIBRARY PROGRAM PUBLICATION

The publications that contain information about the library subprograms available with the operating system are:

IBM System/360 Operating System: FORTRAN IV (E) Library Subprograms, Form C28-6596

IBM System/360 Operating System: PL/I Subroutine Library Computational Subroutines, Form C28-6590

CONTROL PROGRAM SERVICES

The assembler language macro-instructions used for calling upon the services of the supervisor and data management routines of the operating system are described in IBM System/360 Operating System: Control Program Services, Form C28-6541. This publication also provides directions for using test translator (TESTRAN) macro-instructions, which permit problem programs to be tested during their execution.

BASIC TELECOMMUNICATIONS ACCESS METHOD

Information needed to use the basic telecommunications access method is provided in IBM System/360 Operating System: Basic Telecommunications Access Method, Form C30-2001. In addition to general information about telecommunications, this publication describes the macro-instructions provided for the applications programmer and includes a sample program.

MESSAGES, COMPLETION CODES, AND STORAGE DUMPS

The messages produced by all parts of the operating system are explained in IBM System/360 Operating System: Messages, Completion Codes, and Storage Dumps, Form C28-6631. This publication also interprets completion codes, which indicate the reasons for abnormally terminating tasks. Finally, the format of storage dumps associated with abnormal termination are described.

JOB CONTROL LANGUAGE

The statements used to define how jobs are to be processed are described in IBM System/360 Operating System: Job Control Language, Form C28-6539. This publication describes the purposes of the control statements and how they are coded. Examples are provided of typical sequences of the control statements.

Detailed flowcharts illustrating how to code job control statements are provided in the publication IBM System/360 Operaing System: Job Control Language Charts, Form C28-6632.

SYSTEM CONTROL BLOCKS

Descriptions of the fields and illustrations of the formats of the major control blocks of the control program are provided in IBM System/360 Operating System: System Control Blocks, Form C28-6628.

LINKAGE EDITOR

The purpose and use of the linkage editor is described in IBM System/360 Operating System: Linkage Editor, Form C28-6538. This publication describes and gives examples of use of the linkage editor to combine separately compiled parts of programs into a single, executable whole and to divide programs too large to fit into available main storage space so that the parts can be executed separately. The functions performed automatically by the linkage editor to prepare programs for loading are also described.

SORT/MERGE

A description of the sort/merge program is provided in IBM System/360 Operating System: Sort/Merge, Form C28-6543. This publication explains how the generalized sorting and merging capabilities can be tailored to the needs of a particular application. Directions are provided for control statement preparation and input/output device assignment. The publication also contains descriptions of program-generated messages from and timing estimates for the sort/merge program.

UTILITIES

The utility programs and their use are described in IBM System/360 Operating System: Utilities, Form C28-6586. This publication describes and gives examples of how to use the utilities that are provided for organizing and maintaining operating system data.

GRAPHIC PROGRAMMING SERVICES

Descriptions of the programming aids supplied for the graphic devices that may be used with the system are provided in:

IBM System/360 Operating System: Express Graphic Programming Services for IBM 2260 Display Station (Local Attachment), Form C27-6925

IBM System/360 Operating System: Express Graphic Programming Services for IBM 2250 Display Unit, Model 1, Form C27-6921

IBM System/360 Operating System: Basic Graphic Programming Services for IBM 2260 Display Station (Local Attachment), Form C27-6912

IBM System/360 Operating System: Basic Graphic Programming Services for 2250 Display Unit, Form C27-6909

Detailed descriptions of the macro-instructions used to display, modify, and retrieve data in graphic form are included.

SYSTEM PROGRAMMER PUBLICATIONS

The following publications are directed to the system programmer. They provide information to assist him in generating and maintaining an operating system, in adding additional features to it, and in preparing for telecommunications applications.

SYSTEM GENERATION

The procedure used to generate the operating system is described in IBM System/360 Operating System: System Generation, Form C28-6554. This publication is directed to the system programmer assigned to code the system generation run and subsequent system update runs. It describes the system generation language (represented by macro-instructions) that is used to define the system.

STARTER SYSTEM

Information needed to prepare the starter operating system package for use in generating an operating system is provided in IBM System/360 Operating System: Starter Operating System Guide, Form C28-6630. This publication describes the contents of the starter system package, the machine configuration required for its use, initialization procedures, preparation for system generation, and sample programs for testing the generated system.

MAINTENANCE

The procedures for updating the program libraries that constitute the operating system are described in IBM System/360 Operating System: Maintenance Program, Form C27-6918. It describes the capabilities and use of the programming support provided by IBM to update the operating system libraries.

SYSTEM PROGRAMMER'S GUIDE

Information on changing and extending the operating system is contained in IBM System/360 Operating System: System Programmer's Guide, Form C28-6550. This publication is directed primarily to the system programmer who has the responsibility of maintaining the system and incorporating new features into it. It also describes how to perform functions or use facilities that are normally performed or used only by the system programmer.

STORAGE ESTIMATES

The amount of main storage required for the operating system is given in IBM System/360 Operating System: Storage Estimates, Form C28-6551. This publication is directed to the technical manager who is responsible for selecting which optional facilities of the system are required and desired. It contains information for use in estimating the main and auxiliary storage requirements of any configuration of the IBM System/360 Operating System. The publication also includes brief descriptions of control program features.

TELECOMMUNICATIONS

The use of the telecommunication facilities is described in IBM System/360 Operating System: Telecommunications, Form C28-6553. This publication is directed to the system programmer. It contains information on planning and implementing a system for remote message handling and/or processing under the operating system. The publication also describes how to use the control program for performing the input/output operations for a data communications system.

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

CUT ALONG LINE

READER'S COMMENTS

Title: IBM System/360 Operating System
Introduction

Form: C28-6534-1

Is the material:	Yes	No
Easy to Read?	___	___
Well organized?	___	___
Complete?	___	___
Well illustrated?	___	___
Accurate?	___	___
Suitable for its intended audience?	___	___

How did you use this publication?

___ As an introduction to the subject

___ For additional knowledge

Other ___________________

fold

Please check the items that describe your position:

___ Customer personnel
___ IBM personnel
___ Manager
___ Systems Analyst
___ Operator
___ Programmer
___ Customer Engineer
___ Instructor
___ Sales Representative
___ Systems Engineer
___ Trainee
Other ___________

Please check specific criticism(s), give page number(s),and explain below:

___ Clarification on page(s)
___ Addition on page(s)
___ Deletion on page(s)
___ Error on page(s)

Explanation:

fold

Name ___________________
Company ___________________
Address ___________________
City ___________________
State ___________ Zip Code ______

FOLD ON TWO LINES,STAPLE AND MAIL
No Postage Necessary if Mailed in U.S.A.

staple staple

fold fold

FIRST CLASS
PERMIT NO. 81

POUGHKEEPSIE, N.Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

IBM CORPORATION
P.O. BOX 390
POUGHKEEPSIE, N. Y. 12602

ATTN: PROGRAMMING SYSTEMS PUBLICATIONS
DEPT. D58

fold fold

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

staple