File No. S360-20
Form C28-6510-0

IBM Systems Reference Library

IBM System/360 Programming Systems Summary

This publication describes the general function and application of the operating system and the special support system of the IBM System/360.

The operating system consists of a comprehensive set of commercial and scientific programming aids operating under the supervisory control and coordination of an integrated set of control programs. The programming aids and control programs of the operating system can be integrated in various combinations with selected processing, storage, and input/output facilities of the IBM Computing System/360 to form a balanced total system (the IBM System/360) for a particular range of applications. The operating system can be used to perform, individually or concurrently, several types of processing, such as stacked job and remote message processing, on one or more processing units. The programming aids include assemblers having macro capabilities; program compilers, including compilers that are capable of compiling source programs written in FORTRAN, COBOL, or a New Programming Language; a program tester for debugging programs; input/output control programs for use in remote message processing, as well as sequential and direct-access file processing; a report program generator; a generalized sort/merge program; and various utility programs.

The special support system consists of a set of related control programs and programming aids that are designed to provide early delivery of programming support for small card, tape, or disk configurations of the IBM Computing System/360. The control programs and programming aids in this system include input/output control programs; assemblers; a FORTRAN compiler; a New Programming Language compiler; a report program generator; sort/merge programs; utility programs, including program load and storage dump programs; and a 7090/7094 support package that enables programs written for the IBM System/360 in basic assembler language to be assembled and executed on the IBM 7090/7094 Data Processing System.

PREFACE

The purpose of this publication is to serve as a general introduction to the operating system and the special support system of the IBM System/360. The publication is divided into two main sections. The first section describes the operating system and the second section describes the special support system. More comprehensive information on the operation characteristics of these systems may be obtained by reference to the following publications:

Operating System

IBM System/360 Operating System Control Programs, Form C28-6512

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

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

IBM System/360 Operating System FORTRAN Language, Form C28-6515

IBM System/360 Operating System Report Program Generator, Form C28-6517

IBM System/360 Operating System Sort/Merge Program, Form C28-6518

IBM System/360 Operating System Utility Programs, Form C28-6519

Special Support System

IBM System/360 Special Support Basic Assembler Language, Form C28-6503

IBM System/360 Special Support FORTRAN Language, Form C28-6504

IBM System/360 Special Support Utility Programs, Form C28-6505

IBM 7090/7094 Support Package for IBM System/360, Form C28-6501

Because of the many combinations of programming and computing facilities that are possible with the IBM System/360, no attempt is made in this summary to relate the operating system and special support facilities to detailed machine requirements.

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 with a 120-character print chain containing upper and lower case letters, special characters, and numerals.

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

A form has been provided at the back of this publication for readers' comments. If the form has been detached, comments may be directed to an IBM Systems Engineer or addressed to the IBM Corporation, Programming Systems Publications, Dept. D58, PO Box 390, Poughkeepsie, N.Y. 12602.

CONTENTS

OPERATING SYSTEM

Introduction

The operating system consists of a comprehensive set of commercial and scientific programming aids operating under supervisory control and coordination of an integrated set of control programs. The number and types of control programs and programming aids employed in the operating system vary depending upon the short- and long-term requirements at a particular installation. Each operating system consists of a selection of control programs and programming aids that are closely integrated with a selection of processing, storage, and input/output facilities to form a balanced total system designed to satisfy specific requirements. Since data-processing requirements may vary from day to day or hour to hour, the operating system is designed to be easily modified or adjusted to reflect short-term as well as long-term changes in requirements.

As data processing requirements at an installation increase, the operating system, as well as the computing system, can easily be expanded in terms of both performance and application. The ability to expand 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.

The operating system is designed for use throughout the world. For example, it can handle sterling and other currency conventions and it can easily be modified to use national character sets and to communicate with the operator and programmer in national 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 world-wide in scope.

A wide selection of advanced programming aids is available for inclusion in the operating system. These are designed to reduce the time, training, expense, and manpower required to prepare and execute efficient production programs on the Computing System/360. The programming aids may be used singly or in combination to perform a variety of functions such as the following:

1. Compiling machine-language object programs from source programs written in a form of mathematical notation (FORTRAN), or a concise form of the English language (COBOL), or a New Programming Language having features of both FORTRAN and COBOL.

2. Assembling object programs from source programs written in a flexible, easy-to-use, symbolic language.

3. Dynamically testing object programs for errors by providing the programmer with a choice of easily interpreted storage displays that he can specify using symbols and definitions that were used in the original source program.

4. Generating programs for preparing reports from data files.

5. Loading object programs into main storage; if necessary, combining program segments that were individually compiled or assembled, using the same or different source languages, and overlaying executed portions of a program with portions yet to be executed.

6. Performing service and utility operations such as sorting data records, or transferring data from one input/output device to another.

7. Controlling and coordinating all input/output operations thereby relieving the programmer of the task of writing complex input/output routines.

8. Controlling the assignment of data files to input/output devices.

9. Providing a variety of services during the execution of a program that are either specifically requested by the program or automatically provided when a contingency occurs, such as the detection of an error.

10. Modifying or expanding the operating system itself to reflect changes in requirements, to incorporate design improvements, or to incorporate user programs.

Depending on the requirements at a particular installation, the operating system can be tailored and used to control individually or concurrently several types of data processing including the following:

Stacked Job Processing, in which the system is used to perform a continuous series of unrelated jobs with little or no human intervention.

Remote Message Processing, in which the data processing and programming facilities of the IBM System/360 are extended to remote locations thereby enabling the system to become a direct and integral part of the activity that it supports.

Peripheral Processing, in which the system is used to perform supporting functions such as the preparation of a common input file of jobs and the conversion of data from one storage medium or input/output device to another.

Direct-Access-File Transaction Processing, in which a series of unsorted transactions are processed against a master file located in direct-access storage.

To take full advantage of the inherent speed, capacity, and flexibility of the Computing System/360, the operating system is designed to schedule and control several related or unrelated programs or routines running concurrently. These may consist of user programs as well as portions of the operating system itself and may involve the same or different types of processing. By efficiently allocating the available facilities of the system to more than one program and by switching control from one program to another as delays are encountered while waiting for occurrences, such as the completion of an input/output operation, the operating system helps to ensure that as much of the total system as possible is kept in productive operation. Using this technique, the slack time in one program is taken up by another and the total volume of work that is performed over a period of time is significantly increased.

A System Tailored to Requirements

The operating system is actually a composite of many programs which, like the physical facilities of the Computing System/360, can be united in a variety of combinations because they adhere to commonly established conventions for intercommunication and control. When an operating system is generated for a particular application only those control programs and programming aids that are actually required need be included in the system. In addition, key parts of the operating system contain built-in parameters that may be adjusted when the system is generated to tailor it to a specific computing system configuration and to specific requirements.

The programs that constitute an operating system for an installation are stored in a system library located in auxiliary storage. The system library may be divided among several physical units in order to decrease the time

required to gain access to different parts of the system. Parts of the operating system, after being brought into main storage, remain there over extended periods of time to ensure continuous, coordinated operation of the system and to monitor requests for services. Other parts of the system are brought into main storage from the systems library for limited periods of time to perform specific functions, such as loading a program.

After an operating system is generated and placed in use, it may be used to modify itself from time to time in order to satisfy current requirements. For example, if a programming aid, such as a particular program assembler, is not required for some time, it may be removed from the system to conserve space in the system library. In some cases, several programming aids of the same general type, but having different features, storage requirements, and performance characteristics, are available for inclusion in the system. Several of these may be included in the operating system at the same time so that the applications programmer can select from several programming aids of a given type the one or more having the combination of performance characteristics and features best suited to process his particular job.

Where applicable, the programming aids of the system are designed in such a way that the programmer can, by means of optional control statements, cause key operating characteristics to be temporarily modified to suit the requirements of a particular job. However, in the absence of optional control statements from the programmer, the programming aids normally operate in a mode designed to satisfy the requirements of a majority of the users thereby minimizing the number of statements that must be specified by the programmer. Some programming aids, such as the generalized sort/merge program, are designed so the programmer can, if necessary, easily incorporate additional coding or replace coding to satisfy special requirements.

A programmer may incorporate his own programs and data as part of the operating system for short or long periods of time depending on the frequency at which they are used. Such programs may be in the nature of programming aids designed for a specific installation or they may be in the nature of production programs such as a payroll program. Production (user) programs and programming aids, such as assemblers or compilers, are in fact equivalent in that they are stored in the same format in the system library and are written using the same conventions for communicating with one another and with other parts of the operating system. Both are stored in the system library in a format that enables them either to be loaded in preassigned areas of main storage or to be reassigned by the operating system to other areas of main storage. A problem program or programming aid may be incorporated in the operating system for a single job or it may remain a part of the system for an extended period of time and be used by many different jobs.

Both the operating system and the computing system are designed to take advantage of future developments and improvements in computer technology. Conventions for intercommunication and control that are established in the design of the operating system will enable it to be adapted to new techniques and equipment and thereby grow in performance and application.

Advanced Programming Facilities

The operating system provides the programmer with a variety of programming aids which he may use singly or in combination to process a particular job. These programming aids may be supplemented in the future by others supplied by users of the IBM System/360 or by IBM. Each of the several types of programming aids supplied with the initial system is described briefly below.

FORTRAN Compilers

FORTRAN compilers are provided 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 which was developed and refined over a period of years through the combined efforts of IBM and its customers. It closely resembles the language of mathematics and enables engineers and scientists to define problems in a familiar, easy-to-use notation. The language, together with its compiler, relieves the programmer of the detailed work involved in programming problem solutions and thereby reduces the training and time required to produce efficient workable programs.

New Programming Language Compiler

Compilers can be included in the operating system for use in compiling object programs from source programs written in a New Programming Language which has some features that are characteristic of FORTRAN, and which also incorporates some of the best features of other languages such as string manipulation, data structures, dynamic storage allocation, asynchronous operations, and extensive editing capabilities.

This language is designed to take advantage of recent developments in computer technology and to provide the programmer with a flexible problem-oriented language for programming problems that can best be solved using a combination of scientific and commercial computing techniques. It is designed to be particularly useful for the increasing number of semicommercial, semiscientific applications such as information retrieval and command and control applications.

COBOL Compiler

The COBOL compiler is used to compile efficient production programs from source programs written in COBOL. COBOL (COmmon Business Oriented Language) is a concise well-defined language based on English that provides a convenient method of producing programs to solve commercial data-processing problems. Because it is based on English, COBOL is easy to learn and use. This is illustrated by the following typical COBOL sentences:

```
SUBTRACT DEDUCTIONS FROM GROSS GIVING NET.

PERFORM TAX-CALCULATIONS.

WRITE MONTHLY-STATEMENT.
```

COBOL is a widely used language that was developed as a cooperative effort by a number of computer manufacturers and users. IBM System/360 COBOL contains the usual COBOL facilities and, in addition, has been expanded to include the following:

Report Writer Facility

Sort Facility

Source Program Library

Mass Storage Facility

Tele-Communication Facility

Assembler

An assembler is available for use in the operating system in assembling object programs from source programs written in a flexible but easy-to-use symbolic language. The assembler language is a versatile machine-oriented language that can be used for a variety of applications, both commercial and scientific. A number of facilities for assisting the programmer are provided by the assembler. These include macro facilities as well as facilities for defining constants, for defining data-storage areas, for referring to files and storage locations symbolically, and for using literals.

Program Tester

A program tester is available that enables a program or part of a program to be loaded and dynamically and selectively tested in accordance with simple and concise specifications expressed in terms of symbols and definitions used in the original source program. A variety of testing and monitoring facilities are provided by the program tester including file and storage display facilities designed to simplify the analysis of programming errors.

Report Program Generator

The Report Program Generator provides the programmer with an efficient, easy-to-use facility for generating object programs which, in turn, are used to produce reports from data files. 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 files.

Loader

The loader is used to load an object program into an available area of main storage specified by the operating system. During the loading process any subroutines from the system library that were explicitly called for or implied in the source program are combined and loaded with the object program. If necessary, the loader combines program segments that were individually compiled or assembled, using the same or different source languages, and overlays executed portions of a program with portions yet to be executed.

Utility Programs

Utility programs are provided in the operating system for efficiently performing a variety of operations involving the transfer of data from one storage medium or input/output device to another. These programs are available for use either by the operator or by other programs.

Sort/Merge Program

The sort/merge program is a generalized program that can be used to sort and/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.

Input/Output Control

Input/output control programs are provided in the operating system which relieve the programmer of the task of writing complex input/output routines by automatically performing functions such as the blocking and unblocking of data records, the overlapping of processing with input and output, and the preparation and checking of labels. Input/output control programs are available for use in performing the following types of processing:

1. Processing, in sequential order, either logical or physical records on magnetic-tape or direct-access storage.

2. Processing in nonsequential order records in direct-access storage.

3. Processing messages received from remote locations at unpredictable intervals of time.

To ensure efficient and coordinated operation of the input/output resources of the system, some of the input/output control routines remain in main storage over extended periods of time. Other routines are loaded from the system library with the problem program. Only those routines that are actually required by the program are loaded.

Program Execution Control

A variety of services may be provided by the operating system during the execution of a program. These include services that are specifically requested by a program, such as performing a storage dump, or services that are automatically provided when a contingency occurs, such as diagnosing or attempting a recovery from an error when it occurs. Among the services that may be provided by the operating system are the following:

1. Providing the time of day or other timing services such as keeping track of a time interval during which or at which a particular operation is to be performed.

2. Providing a snap or post-mortem storage dump of all or part of a program.

3. Providing standard procedures for diagnosing and attempting to recover from error conditions and other conditions such as a floating-point overflow.

4. Providing accounting information on the use of machine and programming facilities of the System/360.

5. Keeping a log of errors.

6. Providing a checkpoint recording of a program when computing system facilities must be reallocated to higher priority programs, and providing means for restarting such programs using the checkpoint recording.

7. Providing means for communicating with the operator.

System Librarian

The system librarian may be used to perform the following functions:

1. To assist in generating an operating system tailored to the requirements of an installation.

2. To change the operating system, after it is generated, to reflect changes in requirements, including the addition and deletion of programs in the system library.

3. To modify the operating system to reflect design changes and improvements.

Stacked Job Processing

The operating system can be used to process a continuous series of unrelated 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, more efficiently, and are less subject to human error. As a result, turnaround time, the interval between the time a programmer submits a job for processing and the time he receives results, can be significantly reduced.

In programming a job, the programmer may employ any logical combination of the many programming aids within the operating system. The programmer, in effect, controls and directs the processing of his job from his desk by inserting the proper control cards in his job deck. A job may take many forms. It may be a source program written in FORTRAN language that is to be compiled and executed, or it may be a complex commercial job having several segments involving a number of different programming aids and the passing of files from one segment to another.

Before a particular job deck is processed, it is combined with other job decks to form a single input file of unrelated jobs. To ensure more efficient processing, the cards in the job file are normally read, blocked, and copied on magnetic tape or direct-access storage preparatory to the actual processing of the jobs. This may be done either off-line on a Computing System/360 functioning as an auxiliary to a larger System/360 that will later process the jobs or it may be done on-line, concurrently with job processing or other types of processing. Once the input file of jobs is prepared, the jobs can be processed efficiently by the operating system without long, costly setup delays between jobs or job segments.

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

A programmer may place input data for an object program in the common input file immediately following his job deck. The input data can either be loaded with the program or be called into storage by the program as it is required. Similarly, output data produced by the program can be placed in the common job output file. As a result of using the common system files for data input and output, no setup delays are incurred and input/output units that would otherwise be required are available for other purposes. Since the common input/output files are pre-established by the operating system, the programmer is relieved of defining the nature of the files.

If necessary, a programmer may define other files and specify that each be assigned to a specific type, or one of several types, of input/output or direct-access storage devices. Each file is referred to symbolically by the programmer and the actual assignment of a file to a device is performed by the operating system. Therefore, the programmer need not be aware of the specific input/output configuration of the system, and a particular program can be executed on systems with different input/output configurations, provided enough devices of the specified types are available in a system.

The operator of the system, for the most part, performs relatively routine functions, such as loading and unloading tape reels. Normally he is told exactly what to do and when by means of a printout or display from the operating system. If the operating system cannot complete a job or job segment because of a programming error, it automatically skips to the next job or job segment 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 direct the operating system to institute a new type of processing, such as processing messages from remote terminals, that is to be performed concurrently with the stacked job processing.

Remote Message Processing

The operating system can be used to process messages received from remote locations by way of communication lines and tele-communication equipment. Remote message processing is, in effect, an extension of the full power of the data-processing and programming facilities of the IBM System/360 to remote locations. A message received from a remote location may be in the nature of 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.

By extending the services of the system, via communication lines, directly to the user, the turnaround or response time of the system is reduced from hours to 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 centrally control 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 other similar transactions, all within a few seconds.

The system can also be applied to control similar activities in other industries such as the insurance and sales industries. These activities may be carried out within a single building, such as a department store, or, like an insurance activity, be nationwide or even worldwide in scope. The system may also be applied to directly control automatic processes such as are found in the oil, steel, and papermaking industries. In the fields of education, engineering, and research, the system may be used to control experiments taking place at remote laboratories or to extend the use of advanced programming and computing facilities to the desk of the student or engineer.

Remote message processing differs from more conventional types of data processing mainly in the way in which information enters and leaves the system. In short, the techniques for receiving and sending messages differ, but not necessarily the data processing services that are called for by the message. 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. Therefore, the operating system contains special input/output control programs for use in receiving and sending remote messages, as well as control programs for scheduling and establishing priorities for any data-processing services or tasks that are requested.

Peripheral Processing

The operating system can be used to perform service functions such as the preparation of a common input file of jobs, the processing of a common output file produced by a series of jobs, and the conversion of data from one

storage medium or input/output device to another. Because these are basically input/output functions that involve very little actual processing, they are often performed on a peripheral computing system that serves as an auxiliary to a larger system. This is normally done so that the extensive data-processing facilities of the larger system can be more effectively and efficiently employed on jobs that require such facilities. With the IBM System/360, service functions can be performed on a peripheral computing system that serves as an auxiliary to a larger system, or they can be performed on one system concurrently with other types of processing, such as stacked job processing, in a way that ensures that the data-processing facilities of the system are efficiently employed. (Refer to the section Multiprogramming.)

When peripheral and stacked job processing are performed concurrently, it is possible to incorporate in the operating system optional features that are designed to enable the operator to mount files for one job while other jobs are being processed. As job card decks are read on the card reader and the jobs are readied for continuous processing by placing them in auxiliary storage, the operating system can examine each job to determine its input/output requirements. By thus examining and keeping a running account of future input/output requirements, the operating system can reassign an input/output device to a new job as soon as it is released from a previous job and at the same time provide the operator with any necessary instructions for replacing files on input/output devices. As an optional feature, the operating system can also change the order in which jobs are performed so that the processing of one or more non-setup jobs, that is, jobs that do not require special input files, precede the processing of setup jobs, thereby providing the operator with additional time to mount files for the setup jobs. These features reduce to a minimum the times during which the system is idle and waiting for the operator to mount files before it can begin a job.

Multiprogramming

One of the major ways in which operational efficiency is achieved in the IBM System/360 is by multiprogramming. Multiprogramming is a process by which several related or unrelated programs or portions of programs are performed concurrently, provided enough processing, storage, and input/output facilities are available. While one program is awaiting an occurrence, such as the completion of an input/output operation or the end of a time interval, control of the processing unit is directed to another program in accordance with a preestablished order or priority. The competition among several programs for the processing, storage, input/output, and programming facilities of the system helps to ensure that as much of the system as possible is kept busy performing useful work as much of the time as possible. As a result, the total throughput of the system, that is, the total volume of work performed by the system during a given interval of time, is significantly increased.

Multiprogramming was a prime consideration in the design of the Computing System/360. The design provides facilities for the efficient allocation, scheduling, and dispatching of processing unit control, storage, and input/output among programs being performed concurrently. Facilities are also provided in the design for protecting one program from destruction or interference by another program.

In addition to improving operational efficiency, multiprogramming has other very important features in that it enables two or more different types of data processing, such as stacked job processing and peripheral processing, to be performed concurrently on the same system, both with one another and with programs of the operating system itself. Thus, a single IBM System/360 can be used to perform concurrently different types of processing that would normally require separate special-purpose systems if they were to be

performed with any degree of efficiency. It also enables a gradual transition from one type of processing to another. For example, a system may be applied initially to processing jobs and/or transactions that are entered locally. Then, by gradually adding communication facilities, the services of the system could be extended to the point of origin of the jobs or transactions without costly disruptions. 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 or more than one type of processing concurrently. For example, the system could perform stacked job and peripheral processing during most of the day and only perform remote message processing, either individually or concurrently, during business hours.

SPECIAL SUPPORT SYSTEM

Introduction

The special support system consists of a comprehensive set of related control programs and programming aids which are designed for use with small card, tape, or disk configurations of the Computing System/360. Although some of these control programs and programming aids are similar to those provided in the operating system described in the first section of this publication, they differ in that they do not operate under over-all control of supervisory control programs that permit automatic transition from one job or job segment to another.

The special support system contains the following control programs and programming aids:

1. Assemblers for preparing machine-language object programs from source programs written in a flexible, easy-to-use, symbolic language.

2. A FORTRAN IV compiler for compiling object programs from source programs written in a form of mathematical notation.

3. A New Programming Language compiler for compiling object programs written in a new programming language that has many advanced features.

4. A report program generator for generating object programs that prepare reports from data files.

5. Input/output control routines which, when required, are inserted in source programs as they are assembled, compiled, or generated, thereby relieving the programmer of the task of coding complex input/output operations.

6. A set of sort/merge programs for tape and disk configurations.

7. Utility programs, including programs for loading object programs, for dumping the contents of main storage and for transferring and converting data from one input/output or storage medium to another.

8. A 7090/7094 Support Package containing programs that simulate most of the facilities of the Computing System/360 and thereby provide assistance in achieving a smooth transition from other data-processing systems to the IBM System/360.

These control programs and programming aids are designed to reduce the time, expense, training, and manpower required to prepare and execute efficient production programs on the smaller configurations of the IBM Computing System/360. Each is described briefly in the following sections.

Assemblers

Assemblers are provided in the IBM System/360 special support system for assembling machine-language object programs from source programs written in a nonspecialized symbolic language adaptable to general, commercial, or scientific applications. The Basic Assembler is designed for use in card-only or card and magnetic-tape installations.

For larger tape or disk installations, assembler facilities are provided that are similar to those provided for the operating system described in the first section of this publication.

FORTRAN Compiler

A FORTRAN compiler is provided in the special support system for use in compiling object programs from source programs written in FORTRAN language. FORTRAN (FORmula TRANslation) is a specialized language designed for use in programming mathematical, engineering, and scientific problems. It enables the user to write such programs in terms closely resembling those he uses in stating the problems involved. Among the elements that may be used in writing a FORTRAN statement are constants, variables, arithmetic expressions and statements, and relational and logical expressions. The programmer also can use input/output statements; control statements to direct the sequence of operations; and subprogram statements that allow subroutines to be incorporated into the main program.

The special support system FORTRAN is modeled after the FORTRAN IV languages previously implemented on the IBM 1410/7010 systems, and is augmented to take full advantage of the advanced features of the System/360.

The FORTRAN compiler accepts only card input. It translates source program statements into an object program in relocatable format, suitable for execution on a System/360. The compiler produces diagnostic messages pointing out any programming errors in the use of the language. At the user's option, a listing of the source program is provided.

New Programming Language Compiler

A compiler is provided in the special support system for use in compiling object programs from source programs written in a New Programming Language. This language has some features that are characteristic of FORTRAN and also incorporates some of the best features of other languages, such as string manipulation, data structures, and extensive editing capabilities.

The New Programming Language for the special support system provides the facilities of the one for the operating system described in the first section of this publication that are appropriate for card-only configurations. This new language is designed to provide the programmer with a flexible problem-oriented language for programming problems that can best be solved using a combination of scientific and commercial computing techniques.

Report Program Generator

The Report Program Generator provides the programmer with an efficient, easy-to-use facility for generating object programs which, in turn, are used to produce reports from data files. The reports may range from a simple listing of a card deck to a precisely arranged and edited tabulation of calculated data from more than one input file. The input files may be on tape, disk, or cards. The Report Program Generator for the special support system is similar to the one for the operating system described in the first section of this publication.

Input/Output Control Routines

Input/output control routines are provided in the special support system which relieve the programmer of the task of writing complex input/output routines for performing functions such as blocking and unblocking data records, overlapping processing with input/output, preparing and checking labels, and attempting to recover from error conditions. Calls can be made to input/output routines using simple standard calling statements (such as GET, PUT, OPEN, and CLOSE) in the source program. The routines required by a program are inserted in the program as it is being assembled, compiled, or generated.

Other, more basic, routines for use in operating input/output devices are also provided in the special support system. These routines are described in the publication IBM System/360 Special Support Utility Programs, Form C28-6505. These routines can be used to write a magnetic tape record or tape mark, read a tape record, print a line or write a message on a printer, read a card, print a card, rewind a magnetic tape unit, and backspace tape. Each routine includes a procedure to handle errors in data transmission. A special set of routines is also provided for card-only installations; it includes the routines to read and punch a card, and to write a line or a message.

Sort/Merge Programs

A set of Sort/Merge Programs are provided in the system. These can be used to sort and/or merge fixed- or variable-length records in ascending or descending order. The sorting and merging can be performed using magnetic-tape or disk configurations of the Computing System/360.

Utility Programs

Utility Programs are provided in the special support system for performing the following functions:

1. For loading object programs into preassigned areas of main storage.

2. For loading and, if necessary, relocating object programs or portions of object programs.

3. For dumping the contents of all or part of main storage.

4. For efficiently performing a variety of operations involving the transfer of data from one storage medium or input/output device to another. The utility programs that perform these operations are similar to those for the operating system described in the first section of this publication.

IBM 7090/7094 Support Package

A special support system package is provided for use with the IBM 7090/7094 Data-Processing System. The package consists of three programs that enable a user to assemble and test programs written in the IBM System/360 Basic Assembler language. It therefore enables a user to prepare programs for an IBM System/360 prior to its actual delivery.

The programs of the 7090/7094 support package and their functions are as follows:

1. An Assembly Program that assembles programs written in the Basic Assembler language, performs error checking, and produces program listings.

2. A Simulator Program that executes assembled object programs by simulating a System/360 with up to 65K bytes of storage, the 1052 Printer-Keyboard, 1442-2 Card Read Punch, 1443-2 Printer, and 2400 Series Magnetic Tape Units. Storage dump and trace routines, output messages, and other facilities are provided to aid the user in finding and correcting errors within his programs.

3. An IBM 1401 Input Program that accepts symbolic assembler or System/360 machine language programs on punched cards and writes them onto magnetic tape for input to the 7090/7094 Assembly and Simulator programs.

READERS' COMMENTS

IBM System/360 Programming Systems Summary, Form C28-6510-0

Your comments regarding this publication will help us improve future editions. Please comment on the usefulness and readability of the publication, suggest additions and deletions, and list specific errors and omissions.

USEFULNESS AND READABILITY

fold fold

SUGGESTED ADDITIONS AND DELETIONS

CUT ALONG LINE

ERRORS AND OMISSIONS (give page numbers)

fold fold

Name _____

Title or Position _____

Address _____

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

STAPLE

C28-6510-0

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

CUT ALONG LINE

Printed in U.S.A. C28-6510-0

IBM

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

3/64:20M-HP-

STAPLE