File No. S360-36
Form C24-3431-0

IBM Systems Reference Library

IBM Basic Operating System/360 System Control and System Service Programs (16K Tape)

This publication describes the 16K tape resident version of the IBM Basic Operating System/360. The system is a set of control programs and processing programs provided for smaller configurations of the IBM System/360. Using IBM 2400-series magnetic tape units for on-line program residence, Basic Operating System/360 provides stacked-job processing capability, controls all input/output, and provides for continuous operation of all programs run in its environment. Detailed information is given in this publication on these major topics.

1. Operation with the System Control Program:
   a. System Organization
   b. Supervisor Functions
   c. Job Control Program.

2. Using the System Service Programs:
   a. Linkage Editor
   b. Librarian.

The prerequisite for a thorough understanding of this publication is a basic knowledge of System/360 machine concepts. The publications most closely related to this one are:

1. IBM System/360 Principles of Operation, Form A22-6821.

2. IBM Basic Operating System/360: Data Management Concepts (16K Tape), Form C24-3430.

3. IBM Basic Operating System/360: Supervisor and Input/Output Macros (16K Tape), Form C24-3432.

4. IBM Basic Operating System/360: Assembler Language (16K Disk/Tape), Form C24-3414.

PREFACE

The first part of this publication, under the headings Supervisor, Job Control, and IPL Loader, describes the control program for Basic Operating System/360. This part is of interest to anyone using the system, including system analysts, programmers, and machine operators. The functions of the Supervisor are discussed, and the detailed Job Control statement formats are given. Note, however, that the macro instructions used to communicate with the Supervisor are discussed fully in the Supervisor and I/O Marcos publications as listed on the front cover of this publication.

The second part of this publication, under the headings Linkage Editor and Librarian, is of particular interest to the persons responsible for generating and maintaining the resident system. This part describes the Linkage Editor and Librarian programs fully, and gives a general description for generating systems.

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 IBM Programming Publications, Endicott, New York 13764.

The 16K tape resident version of the IBM Basic Operating System/360 is designed to provide operating system capabilities for 16K and larger System/360 configurations that include four or more IBM 2400-series magnetic tape units. Systems above 16K that do not require the expanded functions provided in the larger operating system packages offered by IBM will benefit from this 16K package. The system is tape resident, using IBM 2400-series magnetic tape units for on-line storage of all programs. Depending on the requirements of the particular application, the system can be expanded to include all processing programs used to perform the various jobs of a particular installation, or it can be tailored to a minimum system to control a single program.

The 16K system consists of the following components.

Control Program

The control program constitutes the framework of the Basic Operating System/360. It prepares and controls the execution of all other programs. The components of the control program are:

1. Supervisor. The Supervisor handles all input/output operations, interruption conditions, and other functions for all problem programs. Part of the Supervisor resides in main storage at all times. Processing time is divided between the Supervisor and the program being executed. This is true for the user's programs as well as other IBM-supplied components of the system. Certain functions of the Supervisor are provided by transient routines that remain on tape until needed and which are then loaded into main storage for execution.

2. Job Control. Job Control runs between parts of a job and prepares the system for execution of all other programs. Job Control is loaded by the Supervisor from tape whenever needed.

3. IPL Loader. The IPL Loader loads the Supervisor into main storage when system operation is initiated. The IPL Loader is loaded from tape simply by selecting the address of the tape unit in the load-unit switches on the system console and pressing the load key.

The control program supervises all input/output functions. Required control program input/output units are:

1. System Residence (SYSRES): system residence unit

2. System Reader (SYSRDR): unit used for Job Control statements

3. System Input (SYSIPT): system input unit

4. System Punch (SYSPCH): system output unit

5. System List (SYSLST): system printer unit

6. System Communication (SYSLOG): medium for operator communication.

System Service Programs

The system service programs provide the functions of generating the system, creating and maintaining the library sections, and editing programs on tape before execution. Minimum Systems can be built that do not include the system service programs. Such minimum systems still require tape residence.

The system service programs are:

1. Linkage Editor. All programs are edited on a specified tape by this program. These programs can then be permanently placed in the core image library of the system, requiring only control statements to call them for execution, or they can be stored on tape temporarily, executed, and then overlayed by new programs.

2. Librarian. This is actually a group of programs, used for maintaining the tape libraries and providing printed and punched output from the libraries. Three libraries are used.

   a. Core image library. All permanent programs in the system (IBM-supplied and user programs) are loaded from this library by the System Loader routine of the Supervisor.

   b. Source statement library. This

library is used to store IBM-supplied macro definitions and user-defined source statement routines (such as macro definitions) on the tape built to provide extended program-assembly capability.

c. Relocatable library. This library is used to store object modules which can be used for subsequent linkage with other program modules. A module also can be a complete program.

Processing Programs

All user programs are run within the Basic Operating System/360 environment, using the functions of the control program. Minimum resident systems may consist of only the control program and one or more user programs or, for other types of applications, the control program and the Linkage Editor, with user programs loaded and edited from cards or tape on a specified tape, and then into main storage for execution. A full system may include the user's programs and the following IBM-supplied programs:

1. Language Translators: Assembler, COBOL, FORTRAN, RPG, and PL/I.
2. Sort/Merge.
3. Utilities.
4. Autotest.

SYSTEM CONFIGURATION

This section presents the minimum system configuration required to operate the Basic Operating System/360 and the features in addition to the minimum that can be supported. The system control programs must always be present in order to execute any other programs.

MACHINE REQUIREMENTS

Minimum features required:

16K bytes of main storage.
Standard instruction set. See Note 1.
One I/O Channel (either multiplexor or selector).
One Card Reader (1442, 2501, 2520, or 2540). See Note 2.
One Card Punch (1442, 2520, or 2540). See Note 2.
One Printer (1403, 1404, or 1443). See Note 2.
One 1052 Printer-Keyboard.
One 2400-series magnetic tape drive (9-track) for system residence and three 2400-series magnetic tape drives for language translator functions. An additional tape drive is required for compile-and-execute functions. Except for system residence, 7-track tape units can be used, but the data-convert feature is required.

Note 1: Language translators may require extended instruction sets.

Note 2: One 2400-series magnetic tape drive may be substituted for this device. (7- or 9-track. If 7-track drives are used, the data-convert feature is required, except when substituted for a printer.)

Additional features supported:

Timer Feature.
Simultaneous Read-while-Write Tape Control (2404 or 2804).
Any channel configuration up to one multiplexor channel and six selector channels.
Tape Switching Unit (2816).
Storage Protection Feature.
Additional main storage up to 1024K bytes.

Problem programs can request I/O operations on the following devices:

1. 1442 Card Read Punch
2. 2501 Card Reader
3. 2520 Card Read Punch
4. 2540 Card Read Punch
5. 1403 Printer
6. 1404 Printer (for continuous forms only)
7. 1443 Printer
8. 1445 Printer
9. 1052 Printer-Keyboard (Only one 1052 is supported. It is used for operator communication.)
10. 2401, 2402, 2403, 2404, and 2415 Magnetic Tape Units.

CONTROL STATEMENT CONVENTIONS

The conventions used in this publication to illustrate control statements are as follows.

1. Uppercase letters and punctuation marks (except as described in items 3 through 5 below) represent information that must be coded exactly as shown.

2. Lowercase letters and terms represent information that must be supplied by the programmer.

3. Information contained within brackets [ ] represents an option that can be included or omitted, depending on the requirements of the program.

4. Options contained within braces { } represent alternatives, one of which must be chosen.

5. An ellipsis (a series of three periods) indicates that a variable number of items may be included.

SUPERVISOR

The Supervisor is the control program that operates with problem programs. Part of the Supervisor always resides in main storage. Certain other routines that are used infrequently are kept in the core image library on the resident tape unit and are called into the transient area when needed. The functions performed by the Supervisor are:

1. Storage protection (optional)
2. Interruption handling
3. Channel scheduling
4. Device error recovery
5. Operator communication
6. Program retrieval
7. End-of-job handling
8. Checkpoint
9. Label processing
10. Timer services (optional).

All functions except certain interruption handling (SVC, I/O, and machine check) are available to the problem program by issuing macro instructions. The programmer is not concerned with machine interruption conditions, since these are handled automatically by the Supervisor.

The Supervisor also contains a communication region for holding information useful to problem programs and to the Supervisor itself.

The Supervisor is generated from a set of source statements by way of an Assembler run.

MAIN STORAGE ORGANIZATION

The Supervisor occupies the lower area of main storage. The transient routines are called into the transient area (overlaying the previous routine in the area) and executed when needed. The area occupied by the problem program begins just past the transient area. The main storage map in Figure 1 shows the relationship between the Supervisor and the problem programs.

I/O UNITS CONTROL TABLES

The principal components of the I/O Units Control Tables are the Logical Unit Blocks (LUB) and Physical Unit Blocks (PUB). These tables, defined when the system is generated, are required for channel scheduling, input/output unit assignment and control, and maintenance of miscellaneous information about jobs, such as multiple I/O assignments.

Each LUB is two bytes and represents one logical (symbolic) I/O unit. Each LUB references an entry in the PUB. LUB's corresponding to logical units are ordered according to the logical units they represent. Information concerning the ordering of LUB's is contained in the subsection under Job Control entitled Logical Unit Block (LUB) and Physical Unit Block (PUB). The LUB's are grouped into two classes: system logical units and programmer logical units. The number of programmer logical units is a system generation parameter with a maximum of 245.

Each PUB is eight bytes and represents one physical I/O unit. Contained in each PUB is such information as the channel and unit numbers of the device, the characteristics of the device, references to the channel queue, and indicators used by the Channel Scheduler, Supervisor, and Job Control. The PUB's are ordered by the number of the channel to which the various devices are attached. The number of PUB's is a system generation parameter with a maximum of 255.

COMMUNICATION REGION

The communication region is a 44-byte storage area within the Supervisor region for use by the Supervisor and problem programs. Certain macro instructions are available to allow access to the information contained in this region. Fields in the communication region are addressed relative to the first byte of the region.

The layout of the communication region in the Supervisor is shown in Figure 2 and is described below.

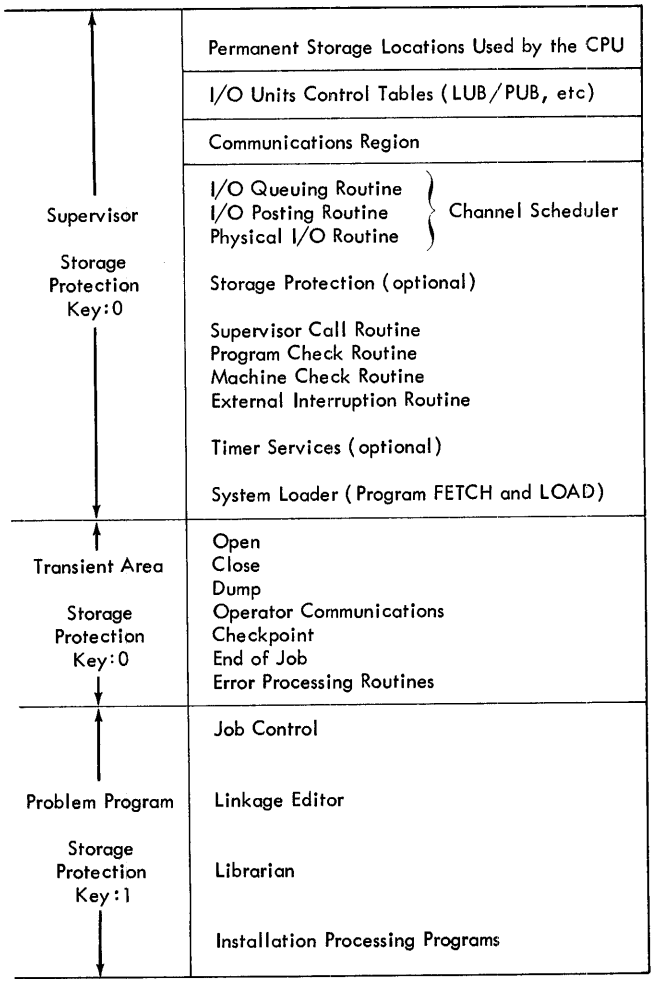


Figure 1. Main Storage Organization

8 bytes	Calendar date. Supplied during IPL procedure or by DATE control statement. Can be used by logical IOCS for dating printed output of the problem program. In one of two forms: mm/dd/yy or dd/mm/yy where mm is month, dd is day, and yy is year.
2 bytes	Address of problem program label area.
2 bytes	Length of problem program label area.
11 bytes	User area (for inter- or intra-program communications). Last 3 bytes set to zero when EXEC control statement is encountered. All 11 bytes set to zero when JOB control statement is encountered.
1 byte	UPSI (user program switch indicators). Set to binary zero when a JOB statement is encountered.
8 bytes	JOB name as found in the JOB control statement.
4 bytes	Address of the uppermost byte of the problem-program area.
4 bytes	Address of the uppermost byte of a phase placed in the problem-program area by the last FETCH or LOAD.

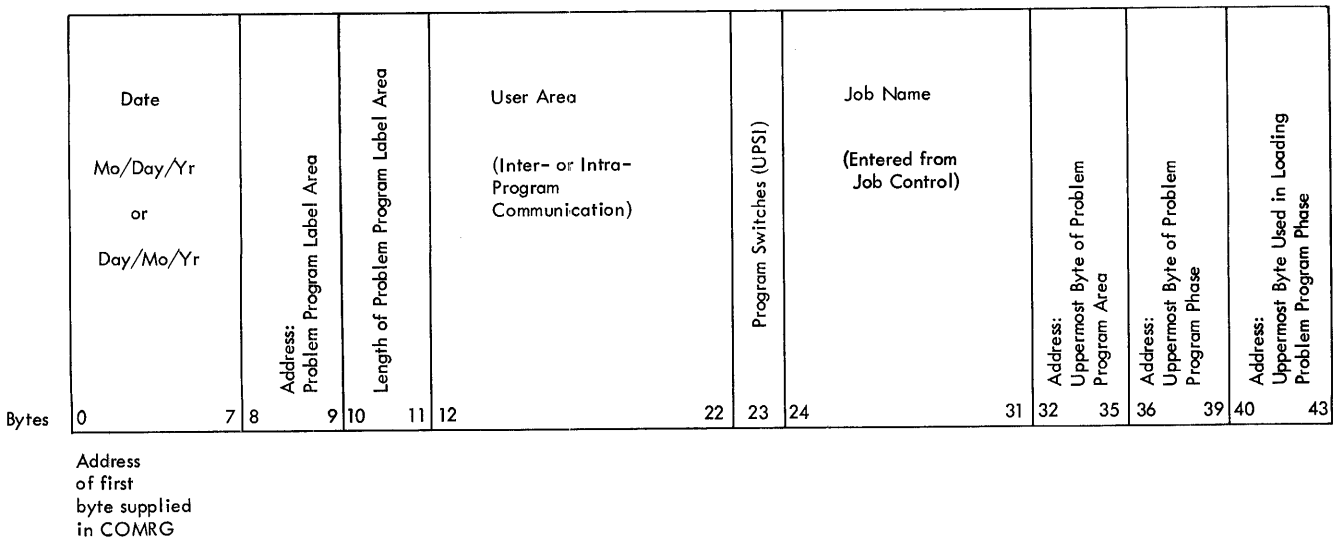


Figure 2. Communication Region in Supervisor

4 bytes	Address of the uppermost byte used in loading any phase of the problem program.

Communication Region Macro Instructions

Macro instructions are provided to allow the problem program to access the communication region. A brief discussion of these macro instructions follows. Details will be found in the Supervisor and I/O Macros publication.

COMRG -- Get address of communication region:
This macro instruction allows the problem program to address information stored in the communication region (obtain date, test switches, etc). The address of the first byte of the region is placed in general register 1.

MVCOM -- Move to communication region: This macro instruction allows the problem program to modify the content of the user area and UPSI (bytes 12 through 23) of the communication region. The operand field of the MVCOM macro instruction contains three operands. The first specifies the first communication region byte to be modified. The second specifies the number of bytes to be inserted. The last specifies the address (or a register containing the address) of the bytes to be inserted.

SUPERVISOR FUNCTIONS

STORAGE PROTECTION

The Supervisor has the responsibility of setting storage protection keys. A storage protection key of 1 is set for the problem program area. A key of 0 is set for the Supervisor and transient areas.

INTERRUPTION HANDLING

An interruption is an automatic transfer of control from any storage location to a predetermined storage location. It can be caused by either a program instruction or a machine condition. The Supervisor automatically handles all interruptions so that the programmer need not be directly concerned with them. In most cases after an interruption is handled, control is returned to the point of interruption as if no break had occurred in the instruction sequence.

There are five kinds of interruptions. They are:

1. Supervisor call
2. External
3. Program check
4. Machine check
5. Input/output.

Figure 3 illustrates the flow of control between the Supervisor and a problem program during an interruption. Control is in the problem program initially. An interruption occurs, transferring control to the Supervisor. The status of the program is saved in the Program Old PSW. Depending on the type and reason for the interruption, control is given to an appropriate handling routine. Upon completion of the routine, the program may be restored to its original condition (via the old PSW). Control is normally given back to the problem program at the point where it was interrupted. The user may have control of program check and external interruptions.

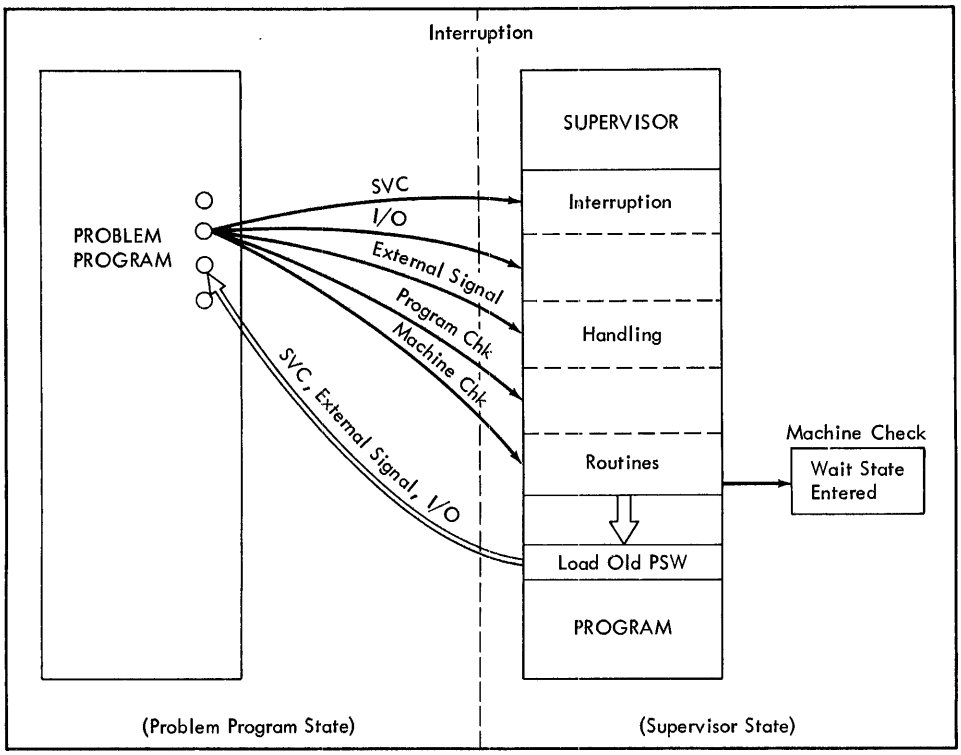


Figure 3. Flow of Control Between Supervisor and Problem Program During an Interruption

Supervisor Call

The supervisor call interruption is caused when the SVC instruction is executed. Certain macros use the SVC (supervisor call) instruction to provide communication between the problem program and the Supervisor. The SVC in each macro has a certain interruption code that indicates to the Supervisor which routine is to be executed.

The macro instructions that allow problem programs to have access to Supervisor functions via an SVC instruction are:

CANCEL To cancel all remaining steps of a job.

CHKPT To cause checkpoints to be taken in a problem program.

CLOSE To close an input/output file.

DUMP To get a printout of main storage and terminate the problem program.

EOJ To call Job Control to prepare the next job step to be run.

EXCP (Execute channel program) To request an I/O operation to be performed by physical IOCS.

EXIT To return to the user's main program after the user's handling of an external or program check interruption.

FETCH To load into main storage from the core image library (or SYS000) a program for execution.

LBRET To return from the problem program to an OPEN, CLOSE, or end-of- of-volume routine

LOAD To load into main storage from the core image library (or SYS000) a phase that is not to be executed immediately.

MVCOM To modify the content of the user area in the communication region.

OPEN To open an input/output file for processing.

SETIME To request the Supervisor to interrupt the execution of a program after a specified period of time has elapsed.

STXIT To establish a linkage from the Supervisor to a user routine (program check or external interruption) or to cancel the use of such a routine.

Each macro instruction generates a supervisor call interruption with a speci-

fic parameter. The interruption routine analyzes the parameter and gives control to another routine for the actual handling of the interruption.

External Interruption

An external interruption can be caused by the timer feature, or by the operator pressing the console interrupt key, or by an external signal.

If an interrupt-key or external-signal interruption occurs, control is immediately given back to the interrupted program unless the user has provided an address of his own routine through an STXIT macro instruction. When this is the case, control is transferred to the address specified.

The timer feature enables the control program to provide three functions:

1. Maintains the time of day which the user can reference at any point within the execution of the problem program.
2. Time-stamps the beginning and end of a job. This information can be used for accounting information and can be printed on the devices assigned to SYSLOG and SYSLST.
3. Enables the user to set the timer for a specific internal of time and to get control at a prespecified address after the time interval has elapsed.

If the presence of the timer feature was not specified when the system was generated, all timer interruptions are ignored and cause control to be returned immediately to the interrupted program.

Four macro instructions are provided for use with external interruptions. These macro instructions are GETIME, SETIME, STXIT, and EXIT. Figure 4 illustrates a typical sequence of events following an external interruption.

GETIME -- Get time of day:
This macro instruction can be used at any point in the execution of a problem program to get the time of day. The value returned to the problem program can be in one of three forms, depending upon the requirements of the user. The

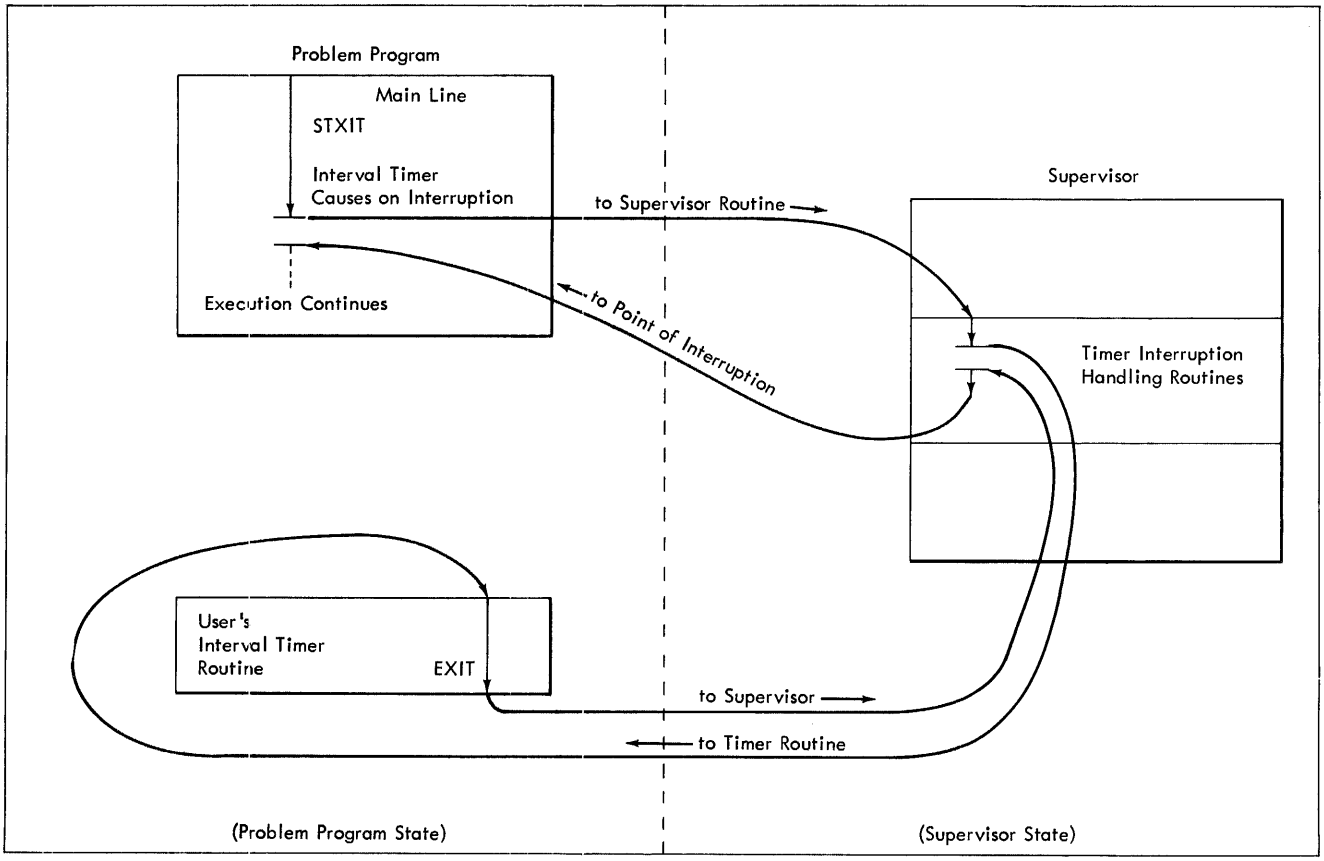


Figure 4. Typical Sequence of Events After a Timer Interruption Due to Use of SETIME

time can be in hours, minutes, and seconds, or it can be a binary integer (in seconds), or it can be in units of 1/300 seconds. This macro instruction can be used only if the timer feature is installed and if its presence is indicated when the system is generated.

SETIME -- Set interval timer:
This macro instruction can be used to request the Supervisor to interrupt the execution of a problem program after a specified time limit has elapsed. This macro instruction can be used only if the timer feature is installed and if its presence is indicated when the system is generated.

STXIT -- Set linkage to user interruption routine:
This macro instruction can be used to establish a linkage from the Supervisor to a user routine. It can also be used to cancel the linkage to such a routine. This macro instruction can be used for external interruptions and program-check interruptions.

EXIT -- Set exit:
This macro instruction is used with the STXIT macro instruction to return from the user's routine to the point of interruption.

A complete description of these macro instructions is supplied in the Supervisor and I/O Macros publication.

Program Check

Each program can select which of the following options is to be taken in the event of a program check.

1. Abort -- The job being executed is terminated and a message output on SYSLOG and SYSLST describes the cause of the termination.

2. Dump and Abort -- This is requested by use of the DUMP option in the OPTION control statement. In addition to a message, all registers and main storage are printed on SYSLST. The job step is then terminated.

3. Transfer to User Routine -- If the address of a subroutine is supplied by the user by way of the STXIT macro instruction, the program-check interruption routine will branch to that subroutine when an appropriate interruption occurs. The user routine can determine the cause of the interruption. Return to the point of interruption is possible by means of the EXIT macro instruction.

Machine Check

A machine-check interruption results from a machine malfunction. The machine-check interruption places the system in the wait state with an identifying code in the main storage address register on the console. The system can be restarted only through an IPL procedure.

A machine-check interruption is simulated by the I/O device error routines by loading the machine check PSW if a detected error condition requires exceptional handling by an IBM Customer Engineer. Operation can be continued if the affected device is not required.

Input/Output Interruptions

An input/output interruption can be caused by:

1. I/O completion -- (Channel End). This is the end of transfer of data into or out of main storage or completion of a control operation.

2. I/O attention -- This results from pressing the request key of the 1052.

3. Device available -- (Device End). A device which was busy or not ready is now available for use.

4. Control unit available -- (Control Unit End). A control unit which was busy is now available for use.

When one of these conditions is detected, control transfers to the Channel Scheduler. (Note that except for posting a program-controlled interruption (PCI) to the user CCB, a program-controlled interruption, occurring as the result of the PCI flag in a CCW, is ignored by the Channel Scheduler.)

CHANNEL SCHEDULER

Complete control of data input/output is accomplished through two kinds of routines: physical and logical. Supervisor input/output control consists of only physical routines (the actual transfer of data between main storage and some I/O device).

These routines make up the Channel Scheduler. The functions it performs are:

1. Schedule I/O requests on each channel (queuing).

2. Start input/output operations.

3. Handle I/O interruptions--normal completion of data transfer, error detection, end-of-file detection, attention (1052).

4. Perform error detection and correction.

5. Detect end-of-job and end-of-job-step control statements on SYSRDR and SYSIPT.

All I/O devices in the System/360 are attached to channels rather than attached directly to the CPU. A channel provides a path for data transfer between the CPU and the I/O device and allows I/O operations to be overlapped with CPU processing and I/O operations on other channels. That is, instructions can be executed simultaneously with data movement in one or more input/output channels. For instance, at a given point in time, one channel may be reading data from a tape, another channel may be writing data on a printer, and a previously read record may be being processed. This is referred to as read/write/compute overlap.

Two types of channels are provided in this system: selector channels and the multiplexor channel. The selector channels allow I/O operations for devices on these channels to be overlapped with CPU processing and I/O operations on other channels. Card, printer, and other low-speed (byte interleave mode) I/O devices can be overlapped with each other, with CPU processing, and with I/O operations on other channels. Thus, greater throughput can be achieved if high-speed devices (tape) are attached to selector channels, and low-speed devices (card and printer) are attached to the multiplexor channel.

Overlapping I/O operations with CPU processing is inherent in the design of the machine and the Channel Scheduler. However, achieving maximum overlapping also partially depends on the problem program. For instance, if overlap is desired in tape operations, the problem program should provide for two I/O areas (or buffers). This allows data to be read into, or written from, one I/O area while records are being processed in the other area. Certain devices, however, have buffers built into the device (1403) and require only one I/O area in main storage to achieve overlap. The use of multiple I/O areas and separate work areas is discussed more fully in the Data Management publication, as listed on the front cover of this publication.

All requests for I/O operations are handled by the Channel Scheduler. When a request is received and the affected channel and device are not busy, the requested operation starts and control passes back to the problem program. If the channel or device is busy, the request is placed at the end of a list (or queue) of I/O requests and the operation is performed as soon as all previous requests have been handled. (Separate queues are maintained for each device.)

The Channel Scheduler also handles all I/O interruptions. If the interruption indicates the normal end of an I/O operation (channel end and no errors), the channel Scheduler posts completion and removes the request from the queue. It then examines the queues for the affected channel. If the queues are empty, control is returned to the problem program at the point of interruption. If instead a request is pending, the Channel Scheduler starts the I/O operation and then returns to the problem program. Requests for devices for which device-end interruptions are outstanding cannot be serviced until the device end is reached. These requests will be bypassed when the Channel Scheduler is selecting an I/O operation to be started. For example, for a 1403, channel end is received as soon as the buffer has been completely loaded (about 4ms), but device end is not received until completion of the print operation (100ms).

The Channel Scheduler detects the following specific status conditions:

1. Wrong Length Record (WLR)

2. End of File (EOF)

3. End of tape

4. Channel and device errors.

Upon detection of the WLR condition, the Channel Scheduler sets an indicator on and supplies the residual count to the problem program. At this point, after posting channel end, the Channel Scheduler reschedules the channel.

Upon detection of an EOF condition, the Channel Scheduler sets an indicator on for the problem program and, as above, reschedules the channel.

If an error is detected, all interruptions remain disabled and the Channel Scheduler passes control to the appropriate device error recovery routine, which takes appropriate action -- retry, operator

intervention, notify problem program, or terminate job.

For certain devices (1052), an I/O operation can be initiated by the operator. To do this, the operator presses the request key on the device. When the Channel Scheduler detects an attention status condition, it passes control to the Supervisor 1052 routine.

A problem program can perform I/O operations in two ways:

1. The problem program can issue physical I/O macro instructions directly.
2. The problem program can use logical IOCS, which in turn issues the physical I/O macro instructions.

Physical I/O Macro Instructions

The physical I/O macro instructions are:

1. EXCP (Execute Channel Program). This macro instruction communicates directly with the Channel Scheduler to request that an I/O operation be started. When the EXCP macro instruction is used, the problem program must supply the appropriate channel program consisting of channel command words (CCW's).
2. WAIT. This macro instruction suspends program operation until an I/O operation (referenced in the WAIT macro instruction) is complete. The problem program must use this macro instruction at the point where processing cannot proceed until the I/O operation is complete. For instance, a problem program may issue the EXCP macro instruction to read a tape block. At the point where the program needs the block for processing, a WAIT macro instruction must be issued. The instructions generated from this macro simply loop, testing a program switch to see if the operation has been completed. The completion of the operation causes an I/O interruption to the Channel Scheduler. Before control is returned to the wait loop, the switch is set to show the completion. Thus, the next time it is tested, the loop is broken and processing continues.
3. CCB (Command Control Block). This declarative macro instruction generates a command control block for each list of CCW's to be executed. The command control block contains information required by the Channel Scheduler to execute the EXCP and WAIT macro instructions. The block is used to pass information between the problem program and the Channel Scheduler, such as status of the operation, action to be taken in the event of an error, etc.

A complete description of these macro instructions is supplied in the Supervisor and I/O Macros publication.

DEVICE ERROR RECOVERY

Each I/O device or class of I/O devices has a unique device error recovery routine. The appropriate routine is entered from the Channel Scheduler upon detection of an error. All these routines have one function in common. That is, an attempt is made to recover from the error. This may be by programming (re-reading tape) or by operator action (2540 not ready).

If recovery is not possible, the following choices are available where applicable.

1. The record in error can be bypassed.
2. An error on an input record can be ignored.
3. The problem program can take action (an exit to a user routine is allowed).
4. The job can be terminated.

Depending on the type of device and on whether logical IOCS is used, some or all of the above options are available. In the absence of any other options, only choice 4 is available.

OPERATOR COMMUNICATION

Communication with the operator is through use of full-text messages issued via the IBM 1052 Printer-Keyboard. Two-way communication is possible: from the system to the operator and from the operator to the system.

The Supervisor permits:

1. Full-text messages to the operator. These messages are either information only or indications of required operator action.
2. Operator-initiated instructions to the Supervisor.
3. Response from the operator to the problem program.

COMMUNICATION TO THE OPERATOR

The control program communicates with the operator by issuing messages. Each message will be preceded by an action indicator that may require a reply from the operator before processing can continue.

The message is printed on the IBM 1052 Printer-Keyboard. If no response is required, an I indicator is included in the message and processing continues. If a reply is required, an action indicator A or D is included in the message. The Supervisor waits until the operator keys in a response.

Each system-to-operator message consists of a four-character message code, a one-character operator action indicator, at least one blank, and the message itself. The action indicator specifies the type of operator action required. The message contains all information pertaining to the operator's decision and/or actions.

The action indicators are as follows.

Indicator		Meaning
A	Action:	The operator must perform a specific manual action before continuing. An example of this is the mounting of a magnetic tape.
D	Decision:	The operator must make a choice between alternate courses of action.
I	Information:	The message does not require immediate operator action. For example, this type of message can be used to indicate the termination of a problem program.

COMMUNICATION FROM THE OPERATOR

The operator may enter a command to the system via the 1052 Printer-Keyboard in any of the following instances.

1. The operator has requested it by pressing the request key.
2. The system has requested operator response.
3. The programmer has requested operator response with a PAUSE statement.

Once a command has been processed, the printer-keyboard is unlocked to permit the issuing of further messages.

Each operator-to-system command consists of two parts. The first part is an operation code of from one to eight alphabetic characters describing the action to be taken. Separated from the operation code by at least one blank are any necessary parameters. The parameters are separated by commas. The command ends with an end-of-block (Ⓑ = Alter Code 5). See End-of-Communication Command.

In order that processing continue, an end-of-communications command consisting of only Ⓑ must be given by the operator following the last command.

With the exceptions of CANCEL and PAUSE, operator-to-system commands are accepted only between job steps. Operator-to-system commands are recognized on SYSRDR.

There are three types of operator-to-system commands.

1. I/O commands
2. Job Control commands
3. Information commands.

I/O Commands

There are six I/O commands: ASSGN (assign logical name), CLOSE (close system output unit), DVCDN (device down), DVCUP (device up), MTC (magnetic tape control), and RESET (reset I/O device assignments).

ASSGN -- Assign Logical Name: The ASSGN command is used to assign a logical I/O unit to a physical device.

Oper-ation	Operand
ASSGN	SYSxxx,address[,X'ss'][,ALT][,TEMP]

The entries in the operand field represent the following.

SYSxxx — The symbolic unit name. It may be one of the following:

SYSRDR
SYSIPT
SYSPCH
SYSLST
SYSLOG
SYSRLB
SYSSLB
SYS000-SYS244

address — Can be expressed as X'cuu', UA, or IGN.
X'cuu' -- Indicates the channel and unit number (in hexadecimal).
c = 0 for multiplexor channel, 1-6 for selector channels 1-6
uu = 00 to FE (0 to 254) in hexadecimal
UA -- Indicates the logical unit is to be unassigned.
IGN -- Indicates the logical unit is to be unassigned and that all program references to the logical device are to be ignored.

X'ss' — Device specifications (used for seven-track tape). The specifications are:

ss	Bytes per Inch	Parity	Translate Feature	Convert Feature
10	200	odd	off	on
20	200	even	off	off
28	200	even	on	off
30	200	odd	off	off
38	200	odd	on	off
50	556	odd	off	on
60	556	even	off	off
68	556	even	on	off
70	556	odd	off	off
78	556	odd	on	off
90	800	odd	off	on
A0	800	even	off	off
A8	800	even	on	off
B0	800	odd	off	off
B8	800	odd	on	off

ALT — Indicates an alternate magnetic tape unit that is used when the capacity of the original assignment is reached.

TEMP — Indicates the assignment for the logical unit will be destroyed by the next JOB statement. Unless this option is taken, the assignment made is carried from job to job.

CLOSE -- Close System Output Unit Command: The CLOSE command is used to close a workfile on the unit designated in the operand field.

Operation	Operand
CLOSE	SYSxxx

The entry SYSxxx can be SYSLST, SYSPCH, or SYS000 to SYS244.

DVCDN -- Device Down Command: The DVCDN command is used to inform the system that a device is no longer physically available for system operations.

Operation	Operand
DVCDN	X'cuu'

The entry X'cuu' is expressed in hexadecimal form, where c is the channel number (0-6) and uu is the unit number, 00-FE (0-254) in hexadecimal.

DVCUP -- Device Up Command: The DVCUP command is used to inform the system that a device is available for system operations after the device has been down.

Operation	Operand
DVCUP	X'cuu'

The entry X'cuu' is expressed in hexadecimal form, where c is the channel number (0-6) and uu is the unit number, 00-FE (0-254) in hexadecimal.

MTC -- Magnetic Tape Control Command: The MTC command is used to control magnetic tape operations. The first entry in the operand field specifies the operation to be performed.

Operation	Operand
MTC	opcode,X'cuu'

The first entry in the operand field can be:

Opcode	Meaning
BSF	Backspace file
BSR	Backspace record
ERG	Erase gap
FSF	Forward space file
FSR	Forward space record
RUN	Rewind and unload
REW	Rewind
WTM	Write tape mark

The second entry X'cuu' is expressed in hexadecimal form, where c is the channel number (0-6) and uu is the unit number, 00-FE (0-254) in hexadecimal.

RESET -- Reset I/O Assignments Command: The RESET command is used to reset I/O assignments to the standard assignments. The standard assignments are those specified when the system was generated plus any

modifications made by the operator via an ASSGN command (without the TEMP option).

Operation	Operand
RESET	blank

The operand field is ignored by the system.

Job Control Commands

There are two Job Control commands: CANCEL (cancel job) and PAUSE (pause).

CANCEL -- Cancel Job Command: The CANCEL command is used to cancel the execution of the current job. The command is acceptable at any time and cancellation is immediate.

Operation	Operand
CANCEL	blank

PAUSE -- Pause Command: The PAUSE command is used to cause Job Control processing to pause between job steps. At that time, the printer-keyboard is unlocked for message input. The end-of-communications indication B causes processing to continue.

Operation	Operand
PAUSE	[any user comment]

The operand of the PAUSE message is not processed by the system. It is used only for operator documentation. A pause command can be issued at any time.

Information Commands

There are five information commands: LISTIO (list I/O assignments), LOG (log Job Control statements), NOLOG (suppress logging), SET (set value), and Ⓑ (end of communications).

LISTIO -- List I/O Assignment Command: The LISTIO command is used to cause the system to print a listing of I/O assignments. The listing appears on the printer-keyboard (SYSLOG).

Operation	Operand
LISTIO	{ SYS, PROG, UA, DOWN, SYSxxx }

If the operand SYS is specified, the listing includes all system logical units. If the operand PROG is specified, the listing includes all programmer logical units. If the operand UA is specified, the listing includes all unassigned physical devices. If the operand DOWN is specified, the listing includes all physical devices that are inoperative. If the operand SYSxxx is specified, where SYSxxx is either a system or programmer logical unit, the listing shows only the assignment for unit SYSxxx.

LOG -- Log Command: The LOG command is used to cause the system to log all Job Control statements on SYSLOG until a NOLOG command is sensed.

Operation	Operand
LOG	blank

The operand field is ignored by the system.

NOLOG -- Suppress Logging Command: The NOLOG command is used to cause the system to suppress the logging of all Job Control statements except JOB, PAUSE, *, and /& until a LOG command is sensed.

Operation	Operand
NOLOG	blank

The operand field is ignored by the system.

SET -- Set Value Command: The SET command is used to initialize the date, clock, and UPSI configuration for system operations. The SET command is used primarily during the IPL procedure.

Operation	Operand
SET	[DATE=n1][,CLOCK=n2][,UPSI=n3] [,LINECT=n4]

The entries in the operand field represent the following.

DATE=n1 — Sets the system date permanently to the specified value. n1 has one of the following formats:

mm/dd/yy
dd/mm/yy

mm specifies the month; dd specifies the day; yy specifies the year. The format to be used is the format that was selected when the system was generated.

CLOCK=n2 — Sets the system clock to the specified value. n2 has the following format.

hh/mm/ss

hh specifies hours (00-23); mm specifies minutes (00-59); ss specifies seconds (00-59).

UPSI=n3 — Sets the bit configuration of the UPSI byte in the communication region. n3 consists of one to eight digits, either 0, 1, or X. Positions containing 0 will be set to 0; positions containing 1 will be set to 1; positions containing X will be unchanged. Unspecified rightmost positions are assumed to be X.

LINECT=n4 — Sets the standard number of lines to be printed on each page of SYSLST. n4 is an integer.

Ⓑ -- End-of-Communication Command: The end-of-communications command must be issued whenever the operator is finished communicating with the system. It causes the communications routine to return control to the mainline job.

Operation	Operand
Ⓑ	Blank

Ⓑ is the end-of-block character, alter code 5.

SYSTEM OPERATION WITHOUT A 1052

When a 1052 is not available on the system, a printer must be assigned to SYSLOG. Messages to the operator are printed on SYSLOG, after which an assumed operator response, where applicable, is taken. In most cases, the assumed response results in the termination of the job. There is no communication from the operator, except for I/O device error routines which require operator-stored response in low main storage. In such cases, a message is printed on the printer assigned to SYSLOG and the device error routines wait until the operator stores his response and presses the console interrupt key. PAUSE statements in the Job Control input stream are ignored.

An operational 1052 is required on the system if external interrupt support is indicated when the system is generated.

In addition to the requirement that SYSLOG be assigned to a printer, SYSRDR and SYSIPT must each be assigned to a card reader (may be the same card reader), SYSPCH must be assigned to a card punch, and SYSLST must be assigned to a printer. If SYSLOG and SYSLST are assigned to the same printer, system-to-operator messages may be embedded within user output.

When no 1052 is available, total throughput in the individual installation will suffer, due to the frequent cancellation of jobs resulting from errors, such as incorrect job setup, I/O assignments, etc. In many instances, such errors could be corrected by the operator via the 1052.

SYSTEM LOADER

The System Loader is a permanently core-resident routine in the Supervisor. It loads all programs run in the Basic Operating System/360 environment, with the exception of the core-resident Supervisor itself.

Programs are loaded into main storage from the core image library if they are cataloged as permanent programs in the core image library, or from SYS000 if the program has been linkage-edited just prior to execution. The FETCH and LOAD macro instructions explicitly name the phase to be loaded.

FETCH Macro Instruction

This macro instruction has the format:

Name	Operation	Operand
[name]	FETCH	phasename[,entryname]

This macro loads the routine (named by phasename) into the problem program area and transfers control to the entry point specified (entryname). If no entry point is specified, control passes to the entry point specified by the Linkage Editor.

LOAD Macro Instruction

This macro instruction has the format:

Name	Operation	Operand
[name]	LOAD	phasename[,address]

This macro instruction is used when a phase (named by phasename) is to be loaded into main storage, but not executed immediately. It can be used to load tables and reference material. After the phase is loaded, control is returned to the calling program with the entry point of the newly loaded phase in register 1. This entry point is specified by the Linkage Editor. If the optional address parameter is specified, then the phase is loaded at that address, disregarding the load point indicated by the Linkage Editor. If the address differs from that provided by the Linkage Editor, no relocation of address constants within the program is provided.

CHECKPOINT/RESTART

When a problem program is expected to run for an extended period of time, provision should be made for taking checkpoint records periodically during the run. The records contain the status of the job and system at the time the records are written. Thus, they provide a means of restarting at some midway point rather than at the beginning of the entire job, if processing must be terminated for any reason prior to the normal end of job. Any programmer logical unit (SYS000-SYS244) can be checkpointed.

For example, a job of higher priority may require immediate processing or some malfunction such as a power failure may occur and cause such an interruption. If checkpoint records are written periodically, operation can be restarted using a set of checkpoint records written prior to the interruption. Therefore, the records contain everything needed to re-initialize the system when processing is restarted.

The Basic Operating System/360 includes routines to take checkpoint records and to restart a job at a given checkpoint. The checkpoint and restart routines are included in the core image library when the system is generated. The CHKPT routine is considered part of the Supervisor and is executed in the transient area. The checkpoint routine is called in response to a CHKPT macro instruction in the problem program. The restart routine is called by Job Control when it reads a RSTRT control statement.

Checkpoint records are written on magnetic tape. Each checkpoint is uniquely identified. When restarting, the RSTRT control statement specifies which checkpoint is to be loaded.

CHKPT Macro Instruction

Checkpoint can be executed as often as desired. The CHKPT macro instruction has the format.

Name	Operation	Operand
[name]	CHKPT	SYSxxx,prgadr,endadr[,point]

where:

SYSxxx is the symbolic name of the unit on which the checkpoint information is to be stored.

prgadr is the restart address.

endadr is uppermost byte of problem program area required to restart the program.

point is the name of an 8-byte field. This parameter is required only if tapes are to be repositioned at restart.

A detailed explanation of the CHKPT macro instruction will be found in the Supervisor and I/O Macros publication.

The restart facility is described in the section on Job Control.

LABEL CHECKING

All label checking is performed by transient routines of the Supervisor. These routines are called into the transient area and executed in response to macro instructions issued by the problem program. Since these routines do not occupy space that can be used by the problem program, there is no need to handle OPEN and CLOSE operations as overlays in special phases.

NORMAL AND ABNORMAL END-OF-JOB HANDLING

When a program reaches the normal end of a job step, issuing the EOJ macro instruction causes the Supervisor to fetch Job Control to begin processing the control statements for the next job or job step.

A special routine of the Supervisor can provide a print-out of main storage in the event of some abnormal end-of-job-step situation. This routine is fetched into the transient area when a dump is required and when DUMP is specified as a standard option at system generation time or when DUMP is specified in the OPTION control statement. The dump routine prints the contents of the registers and main storage from location 0 to the end of the problem program area.

The Job Control program provides job-to-job transition within the Basic Operating System/360. It also is called into main storage to prepare each job step to be run. (One or more programs can be executed within a single job. Each such execution is called a job step.) It performs its functions between job steps and is not present while a problem program is being executed. Job Control is called by:

1. The IPL Loader, to process the first job after an IPL procedure.
2. The Supervisor, at normal end of a job step, or at an abnormal end of job.

A macro instruction, EOJ, is provided to call Job Control for a normal end of a job step.

FUNCTIONS

Job Control performs various functions on the basis of information provided in job control statements. These functions are:

1. Prepare programs for execution.
2. Assign device addresses to symbolic names.
3. Set up fields in the communication region.
4. Edit and store volume and file label information.
5. Prepare for restarting of checkpointed programs.

Job Control clears the problem program area in main storage to binary zero between job steps.

PREPARE PROGRAMS FOR EXECUTION

All programs run in the system are loaded from either the core image library or from SYS000. If a program has been previously cataloged (see Librarian) as a permanent entry in the core image library, Job Control has only to transfer to the system loader to load the program for execution from the core image library. If the program to be executed is on SYS000, it will have had all linkages resolved by the Linkage Editor (see Linkage Editor). Job Control transfers to the system loader to load the program for execution directly from SYS000.

SYMBOLIC INPUT/OUTPUT ASSIGNMENT

Job Control is responsible for assigning physical I/O units. Programs do not reference I/O devices by their actual physical addresses, but rather by symbolic names. The ability to reference an I/O device by a symbolic name rather than a physical address provides advantages to both programmers and machine operators. The symbolic name of a device is chosen by the programmer from a fixed set of symbolic names. He can write a program that is dependent only on the device type and not on the actual device address. At execution time, the operator or programmer determines the actual physical device to be assigned to a given symbolic name. He communicates this to Job Control by a control statement (ASSGN). Job Control associates the physical device with the symbolic name by which it is referenced.

A fixed set of symbolic names is used to reference I/O devices. No other names can be used. They are:

SYSRDR	Card reader or magnetic tape unit used for Job Control statements.
SYSIPT	Card reader or magnetic tape unit used as the input unit for programs.
SYSPCH	Card punch or magnetic tape unit used as the main unit for punched output.
SYSLST	Printer or magnetic tape unit used as the main unit for printed output.
SYSLOG	Printer-keyboard used for operator messages and to log Job Control statements. Can also be assigned to a printer.
SYSRES	System residence tape unit.
SYSSLB	Tape unit used for the source statement library.

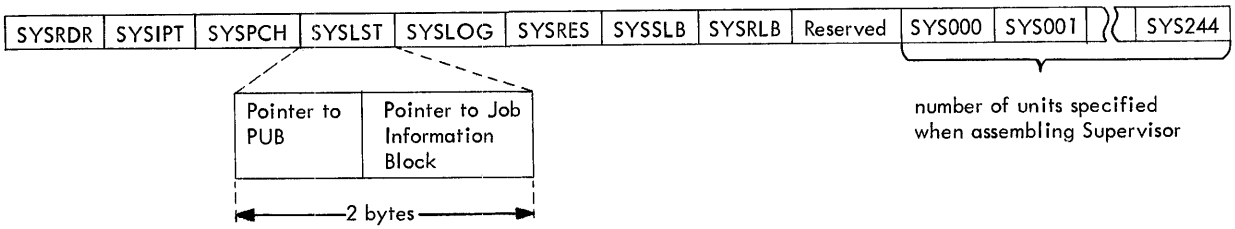


Figure 5. Sequence of LUB's in Device Table

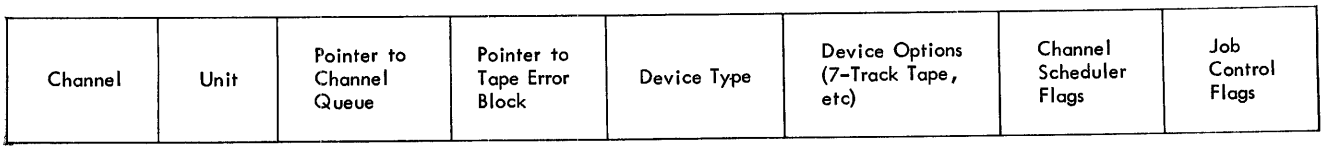


Figure 6. Format of PUB Entry

SYSRLB — Tape unit used for the relocatable library.

SYS000-SYS244 — All other units in the system.

The first eight of the above names, termed system logical units, are used by the system control program and system service programs. Of these nine units, user programs may also use SYSIPT for input, SYSLST and SYPCH for appropriate output, and SYSLOG for operator communication. Normally, SYSRDR and SYSIPT both refer to the same device. SYSRES, SYSSLB, and SYSRLB can all refer to the same device. Any additional devices in the system, termed programmer logical units, are referred to by names ranging consecutively from SYS000 to SYS244, with SYS000 to SYS003 being the minimum provided in any system.

Logical Unit Block (LUB) and Physical Unit Block (PUB)

At system generation time when a Supervisor is assembled, a device table is set up with an entry for each of the symbolic names that will be used in the system. Each entry is called a logical unit block (LUB). The format of the device table is shown in Figure 5. The length of the table depends on the number of devices specified at system generation time. The system LUB's and the first four programmer LUB's are always present.

A physical unit block (PUB) can be associated with each LUB. The format of a PUB is shown in Figure 6. The PUB's are ordered by priority within the channel to which the various devices are attached.

Normally, each symbolic name is assigned a physical device address at the time the Supervisor is assembled. In some cases, a single device may be assigned to two or more symbolic names. An installation can make specific assignments at system generation time and establish these as conventions to be followed by all programmers. By following the conventions, most jobs can be submitted for execution with no ASSGN control statements. Figure 7, for example, shows a typical system configuration. The following conventions might be established for the installation's own programs and for IBM-supplied programs.

1. Control statement input is read from SYSRDR. This device is normally assigned to the same physical unit as SYSIPT. Most of the system programs (language translators, etc) and user programs normally read from SYSIPT.

2. Card putput is punched on SYSPCH.

3. Printed output is on SYSLST.

4. A 1052 is assigned to SYSLOG.

5. The seven tape units are addressed as SYSRES, SYSSLB, SYSRLB, SYS000, SYS001, SYS002, and SYS003. SYSRES is used for system residence. SYSSLB is used for the source statement library. SYSRLB is used for the relocatable library. SYS000 is used by the Linkage Editor when performing its editing functions. SYS001, SYS002, and SYS003 are used by the language translators as work files. Language translators also can output Linkage Editor input on SYS000.

The initial device assignments present after each IPL procedure are those made when the system is generated plus any changes introduced at IPL time. Once the Supervisor is loaded into main storage, reassignments made by the operator on the

1052 become permanent modifications to the existing system assignments unless the operator specifies temporary assignment. Reassignments made by the programmer are reinitialized to the original assignments at the completion of a job.

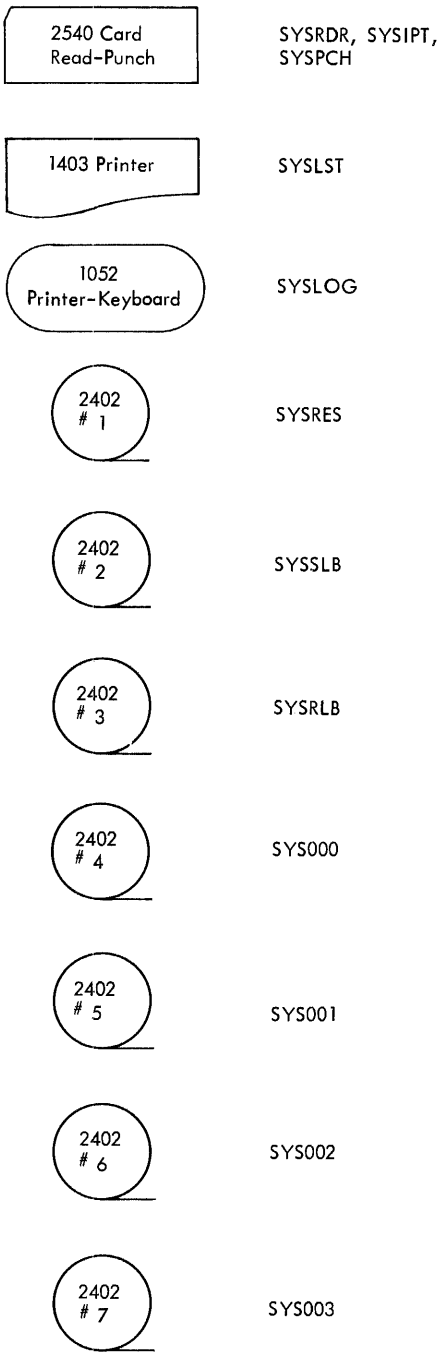


Figure 7. Example of Symbolic Device Assignment

SET UP COMMUNICATION REGION

Job Control takes the following information from control statements and places it in the communication region.

1. Job Name. Taken from the JOB statement. This field can be used by the problem program for accounting purposes.

2. Job Date. Taken from the DATE statement. It can be used by the problem program to date output reports. If the DATE statement is not used, the system uses the date supplied by the operator at IPL time.

3. User Program Switch Indicators. Taken from the UPSI statement. The bit pattern in this byte can be used as switch indicators to specify program options.

EDIT AND STORE LABEL INFORMATION

All volume and file label processing is done during problem program execution. However, label information to be checked against is read from label statements by Job Control and stored in the lower part of the problem program area for subsequent processing. The formats of the label information statements are discussed in this section. See the Data Management publication for a complete discussion of volume and file labels.

RESTARTING PROGRAMS FROM CHECKPOINT

Job Control prepares the system for restarting from a checkpoint by loading the restart program which repositions tape drives, reinitializes the communication region, and stores the information from the RSTRT statement. The restart program handles the actual restarting of the problem program.

JOB CONTROL STATEMENTS

GENERAL CONTROL STATEMENT FORMAT

Certain rules must be followed when filling out control statements. Job Control statements conform to these rules.

1. Name. Two slashes (//) identify the statement as a control statement. They must be in columns 1 and 2. At least one blank immediately follows the second slash. Exception: The end-of-job statement contains /& in columns 1 and 2, the end-of-data-file statement contains /* in columns 1 and 2, and the comment statement contains * in column 1 and blank in column 2.

2. Operation. This describes the type of control statement (the operation to be performed). It can be up to eight characters long. At least one blank follows its last character.

3. Operand. This may be blank or may contain one or more entries separated by commas. The last term must be followed by a blank, unless its last character is in column 71.

All control statements are essentially free form. Information starts in column 1 and cannot extend past column 71. Exception: For the file label statement (TPLAB) information may not be contained entirely in one card image. Any non-blank character present in column 72 specifies that information is continued in the following card image (continuation statement). Information in the continuation statement begins in column 16; columns 1-15 are ignored.

Job Control continuation statements are used only for the label statements (other statements are not checked for this possibility).

Job Control reads from the device identified by the symbolic name SYSRDR. The following statements are recognized:

Operation	Meaning
JOB	Job name
EXEC	Execute program
ASSGN	I/O assignments
RESET	Reset I/O assignments
DATE	Date
UPSI	User program switch indicators
NMTLB	Number of magnetic tape label blocks
VOL	Volume information
TPLAB	Tape file label information
RSTRT	Restart
LISTIO	List I/O assignments
OPTION	Option
PAUSE	Pause
/*	End of data file
/&	End of job
*	Comment

Any statement other than these is recognized as an error. A message is issued so that the programmer can correct the state-

ment in error. Some of the errors recognized are:

1. Invalid symbolic unit name.

2. No space reserved in LUB table for a symbolic unit.

3. Invalid device-type.

4. Invalid length of field.

5. Invalid character.

6. Missing /& statement.

7. A label statement (TPLAB) does not immediately follow its associated volume (VOL) statement.

SEQUENCE OF CONTROL STATEMENTS

The Job Control statements for a specific job always begin with a JOB statement and end with a /& (end of job) statement. A specific job consists of one or more job steps. Each job step is initiated by an EXEC statement. Preceding the EXEC statement are any job control statements necessary to prepare for the execution of the specific job step. The only limitation on the sequence of statements preceding the EXEC statement is that discussed below for the label information statements. The following statements can precede the EXEC statement for a job step.

```
ASSGN
RESET
DATE
UPSI
NMTLB
VOL
TPLAB
LISTIO
OPTION
PAUSE
*
```

The label statements must be in the order:

```
VOL
TPLAB
```

and, must immediately precede the EXEC statement to which they apply.

DESCRIPTION AND FORMAT OF JOB CONTROL STATEMENTS

JOB Statement

This statement indicates the beginning of control information for a job. The JOB statement is in the following format.

```
// JOB jobname
```

jobname — The name of the job. Must be one to eight alphameric characters.

EXEC Statement

The EXEC (execute) control statement must be the last statement processed before a job step is executed. It indicates the end of job control statements for a job step and that execution of a program is to begin. Its format is:

```
// EXEC [progname]
```

progname — Represents the name of the program in the core image library to be executed. The program name can be one to eight alphameric characters. If the program to be executed has just been processed by the Linkage Editor, the operand of the EXEC statement is blank.

ASSGN Statement

When programs are assembled, they use symbolic names to reference I/O devices. At execution time this statement is used to assign a specific device address to the symbolic unit name used. It contains the symbolic unit name and various parameters to describe the physical device. The format is:

```
// ASSGN SYSxxx,deviceaddress[,X'ss'][,ALT]
```

SYSxxx — The symbolic unit name. It may be one of the following.

```
SYSRDR
SYSIPT
SYSPCH
SYSLST
SYSLOG
SYSSLB
SYSRLB
SYS000 to SYS244
```

deviceaddress — Can be expressed as X'cuu', UA, or IGN.

X'cuu' -- channel and unit number (in hexadecimal).
c = 0 for multiplexor channel, 1-6 for selector channels 1-6
uu = 00 to FE (0 to 254) in hexadecimal.

UA -- indicates the logical unit is to be unassigned.

IGN -- indicates the logical device are to be unassigned and that all program references to the logical device are to be ignored.

X'ss' — Device specifications (used for seven-track tape). The specifications are:

ss	Bytes per Inch	Parity	Translate Feature	Convert Feature
10	200	odd	off	on
20	200	even	off	off
28	200	even	on	off
30	200	odd	off	off
38	200	odd	on	off
50	556	odd	off	on
60	556	even	off	off
68	556	even	on	off
70	556	odd	off	off
78	556	odd	on	off
90	800	odd	off	on
A0	800	even	off	off
A8	800	even	on	off
B0	800	odd	off	off
B8	800	odd	on	off

ALT — Indicates an alternate magnetic tape unit that is used when the capacity of the original assignment is reached.

All device assignments made with ASSGN control statements are reset between jobs to the configuration specified when the system was generated plus any modifications that may have been made by the operator at IPL time and between jobs or job steps.

RESET Statement

The RESET statement is used to reset I/O assignments to the standard assignments. Its format is:

```
// RESET
```

The standard assignments are those specified when the system is generated plus any modifications made by the operator via an ASSGN command (as opposed to an ASSGN control statement) without the TEMP option.

DATE Statement

This statement contains a date which is put in the communication region. It is in one of the following formats:

```
// DATE mm/dd/yy
// DATE dd/mm/yy
```

mm = Month (01 to 12)
dd = Day (01 to 31)
yy = Year (00 to 99)

When the DATE statement is used, it applies only to the current job being executed. Job Control does not check the order that the digits appear in the operand. If no DATE statement is used, Job Control supplies the date given in the last SET command.

UPSI Statement

This statement (User Program Switch Indicators) allows the user to set program switches that can be tested much the same as sense switches or lights used on other machines. The UPSI statement has the following format.

```
// UPSI nnnnnnnn
```

The operand consists of one to eight characters of 0, 1, or X. Positions containing 0 will be set to 0. Positions containing 1 will be set to 1. Positions containing X will be unchanged. Unspecified rightmost positions are assumed to be X.

Job Control clears the UPSI byte to zeros before reading control statements for each job. When Job Control reads the UPSI statement, it sets or ignores the bits of the UPSI byte in the communication region. Left to right in the UPSI statement, the digits correspond to bits 0 through 7 in the UPSI byte. Any combination of the eight bits may be tested by problem programs at execution time.

NMTLB Statement

The NMTLB statement directs the Linkage Editor to reserve a portion of the lower problem program area as a label block area. The NMTLB statement must precede the EXEC LNKEDT statement. The format is:

```
// NMTLB nnn
```

where nnn is the decimal number of pairs of VOL/TPLAB statements that will appear immediately prior to the execution of the linkage edited program.

VOL Statement

The volume statement is used when checking or writing standard labels for a tape file. A VOL statement must be used for each file on a multi-file volume. Its format is:

```
// VOL SYSxxx,filename
```

SYSxxx — Symbolic unit name.

filename — File name. This can be one to eight characters and is identical to the symbolic address of the program DTF which identifies the file.

TPLAB Statement

The tape-label statement contains file label information for tape label checking and writing. This statement must immediately follow the volume (VOL) statement. Tape file labels normally require only one statement. This contains fields 3-10 of the standard tape file label. These are the only fields used for checking the label of an input file. When writing output labels, the additional fields (11-13) can be included, if desired, by continuing the TPLAB statement in a continuation statement. Thus, the TPLAB statement may have either of the following two formats.

```
// TPLAB 'label fields 3-10'

// TPLAB 'label fields 3-10     C
         label fields 11-13'
```

'label fields 3-10' — The indicated fields of the standard tape file label are contained just as they appear in the label. This is a 49-byte character string, contained within single quotes. The standard tape file label is shown in Appendix A.

label fields 11-13 — These fields are a 20-character direct continuation of the same character string begun with fields 3-10 (no blanks, quotes, or commas separating). These fields are not required. If intended for an output file, they are written in the corresponding fields of the output label. They are ignored when used for an input file. These fields are never used by the Basic Operating System/360 label processing routines. They can be entered in labels for files that are to be processed by Operating System/360.

RSTRT Statement

A restart facility is available for checkpointed programs. A programmer can use the CHKPT macro instruction in his program to cause checkpoint records to be written. This allows sufficient information to be stored so that program execution can be restarted a specified point. The checkpointed information includes the registers, tape-positioning information, a dump of main storage, and a restart address.

The restart facility allows the programmer to continue execution of an interrupted job at a point other than the beginning. The procedure is to submit a group of job control statements including a restart (RSTRT) statement. The format of the RSTRT statement follows.

```
// RSTRT SYSxxx,nnnn
```

SYSxxx — Symbolic unit name of the device on which the checkpoint records are stored. This unit must have been previously assigned.

nnnn — Identification of the checkpoint record to be used for restart-

ing. This serial number is one to four characters. It corresponds to the checkpoint identification used when the checkpoint was taken. The serial number is supplied by the checkpoint routine.

See the Supervisor and I/O macros publication for further details on the CHKPT macro instruction.

When a checkpoint is taken, the completed checkpoint is noted on SYSLOG. Restarting can be done from any checkpoint record, not just the last. The proper I/O device assignments must precede the RSTRT control statement.

Assignment of input/output devices to symbolic unit names may vary from the initial assignment. Assignments are made for restarting jobs in the same manner as assignments are made for normal jobs.

LISTIO Statement

This statement is used to get a listing of I/O assignments. Its format is:

```
// LISTIO {SYS   }
          {PROG  }
          {UA    }
          {DOWN  }
          {SYSxxx}
```

The listing is output on the device assigned to SYSLST. The listing varies according to the operand used. If the operand SYS is specified, the listing includes all system logical units. If the operand PROG is specified, the listing includes all programmer logical units. If the operand UA is specified, the listing includes all unassigned physical devices. If the operand DOWN is specified, the listing includes all physical devices that are inoperative. If the operand SYSxxx is specified where SYSxxx is either a system or programmer logical unit, the listing shows only the assignment for unit SYSxxx. An example of a listing produced by the LISTIO SYS statement is shown in Figure 8. All channel and unit numbers are represented in hexadecimal.

```
LISTIO SYS

I/O UNITS CH. UNIT

*** SYSTEM ***
 SYSRES     1   00
 SYSLOG     0   1F
 SYSRDR     0   04
 SYSIPT     2   01
 SYSLST     2   02
 SYSPCH     IGN
 SYSSLB     UA
 SYSRLB     1   02
```

Figure 8. Example of LISTIO SYS Output

OPTION Statement

This statement is used to specify one or more of the Job Control options. The format of the OPTION statement is:

```
// OPTION option1[,option2,...]
```

The options that can appear in the operand field are as follows. Selected options can be in any order.

LOG — Causes the listing of all control statements on SYSLST. Control statements are not listed until a LOG option is encountered. Once a LOG option statement is read, logging continues from job-step to job-step until a NOLOG option is encountered or until the JOB control statement is encountered.

NOLOG — Suppresses the listing of all control statements on SYSLST except JOB, PAUSE, *, and LOG option statements until a LOG option is encountered.

DUMP — Causes a dump of the registers and main storage to be output on SYSLST in the case of an abnormal program end (cancel or program check).

NODUMP — Suppresses the DUMP option.

LINK — Indicates the problem is to be linkage-edited immediately after compilation or assembly. When the LINK option is used, the output of the language translators is written on SYS000. A linkage editor run is then performed to linkage-edit the output of the language translators.

NOLINK — Suppresses the LINK option. The language translators can also

	suppress the LINK option if the problem program contains an error that would preclude the successful execution of the problem program.
DECK	Causes language translators to output object modules on SYSPCH. If LINK is specified, the DECK option is ignored.
NODECK	Suppresses the DECK option.
LIST	Causes language translators to write the source module listing on SYSLST.
NOLIST	Suppresses the LIST option.
LISTX	Causes language translators to write the hexadecimal object module listing on SYSLST.
NOLISTX	Suppresses the LISTX option.
SYM	Causes the Assembler to output the symbol table on SYSLST and/or SYSPCH.
NOSYM	Suppresses the SYM option.
XREF	Causes the Assembler to write the symbolic cross-reference list on SYSLST.
NOXREF	Suppresses the XREF option.
ERRS	Causes the language translators to summarize all errors in the source program on SYSLST.
NOERRS	Suppresses the ERRS option.
CATAL	Causes the cataloging of a phase or program in the core image library at the completion of a Linkage Editor run by calling the MAINT librarian program. CATAL also causes the LINK option to be set.
48C	Specifies the 48-character set on SYSIPT.
60C	Specifies the 60-character set on SYSIPT.

The options specified in the OPTION statement remain in effect until a contrary option is encountered or until a JOB control statement is read. In the latter case, the options are reset to the standard that was established when the system was originally generated.

Any assignment for SYS000 after the occurrence of the OPTION statement cancels the LINK and CATAL options. These two options are also cancelled after each occurrence of an EXEC statement with a blank operand.

Note: The LOG and NOLOG control statements defined for Basic Programming Support (8K Tape) and Basic Operating System (8K Disk) are recognized by Job Control as equivalent to the LOG and NOLOG options.

PAUSE Statement

This statement has the following format.

```
// PAUSE [comments]
```

Comments — Optional. Any message to the operator can appear in the operand of a PAUSE statement.

The statement can be used to allow for the operator action between job steps.

The printer-keyboard is unlocked for operator-message input. The end-of-communications indication, (B), causes processing to continue. The PAUSE statement is always printed on SYSLOG. If no 1052 is available, the PAUSE statement is ignored.

/* -- End of Data File Statement

This statement must be the last statement of each input data file on SYSRDR and SYSIPT. Its format is:

```
/* ignored
```

Columns 1 and 2 are the only columns that are checked. /* causes the Channel Scheduler to post the end-of-file indicator in the user's CCB.

/& -- End of Job Statement

This statement must be the last statement of each job. Its format is:

```
/& ignored
```

Columns 1 and 2 contain a slash and an ampersand (12-punch). Upon occurrence of /&, the Channel Scheduler posts an end-of-file indicator in the user's CCB. If the user attempts to read past the /&, the job is terminated.

* -- Comments Statement

```
* any user comments
```

Column 1 contains an asterisk. Column 2 is blank. The remainder of the statement contains any user comments. The content of the comment statement is printed on SYSLOG. If followed by a PAUSE statement, the statement can be used to request operator action.

SYSTEM I/O OPERATIONS

This section describes the opening and closing of magnetic tape devices when assigned to system logical units, and the opening and closing of magnetic tape devices when assigned to four programmer logical units (SYS000 - SYS003) used by the various system components.

OPEN System Files

When the system logical unit SYSRDR, SYSIPT, SYSPCH, or SYSLST is assigned, Job Control checks to see whether the device assignment is to a magnetic tape device. If so, Job Control rewinds the tape and performs an OPEN having the following characteristics.

1. Accepts a leading tape mark for input or output.
2. Accepts a completely unlabeled tape (other than blank tape) for input or output, and writes a leading tape mark if output. Non-standard labels are not recognized.
3. Accepts an IBM-standard label set (including user labels) for input.
4. Accepts a valid IBM-standard label set (including user labels) having a past expiration date for output.
5. Rejects all other volumes by issuing a rewind and unload to the drive. An error message is printed on the 1052 (SYSLOG). The operator can then substitute an acceptable volume and re-ASSGN from the 1052.

CLOSE System Output Files

As the system does not provide for generating multi-reel system output files, the units SYSLST and SYSPCH are closed by the operator when the limit of their capacities is near.

The operator may then mount new volumes on these drives, or if enough drives are available, they may have been previously mounted. The operator then assigns SYSLST and SYSPCH to their respecitve tape drives.

Processing of input resumes with the succeeding job step. The operator should not allow the output files to reach the limit of their capacities.

CONTROL STATEMENT EFFECT ON I/O UNITS

Certain control statements in the Job Control input stream affect the use of system I/O units by Job Control. These statements are as follows.

// JOB. When the JOB statement is encountered, the content of the statement is printed on SYSLOG and SYSLST. If SYSLST is assigned to a printer, it is first skipped to a new page. If SYSLST is assigned to a magnetic tape, a carriage eject character is prefixed to the card image, which is then written on tape. If SYSPCH is assigned to a magnetic tape, two blank card images are written on tape.

/&. When Job Control encounters /& on SYSRDR during normal operation, a check is made to determine whether SYSIPT is assigned to the same physical device. If not, SYSIPT is also advanced to /&.

In the event of an abnormal termination, Job Control advances SYSRDR to the next /&, and proceeds as above.

The user should beware of omitting /&, for protection of one job from errors in the preceding job cannot then be guaranteed.

Job Control has no responsibility for the arrangement of output on any file, except that connected with control statements. Such items as page ejection and line count must be managed by the individual processing program.

WORK FILES USED BY SYSTEM COMPONENTS

The four programmer logical units SYS000, SYS001, SYS002, and SYS003 are used as work files by the various system components (Linkage Editor, Librarian, Language Translators, etc). Each must be assigned to a tape unit. The assignment of SYS000 and SYS003 may be to the same tape unit, however, when the "compile-and-execute" mode of operation is desired, SYS000 and SYS003 must be assigned to separate tape units.

SYS000 is opened only by Job Control and is opened only when OPTION LINK or CATAL is encountered.

The work files SYS001, SYS002, and SYS003 are opened and closed by each system component that uses them. The programmer does not provide label information for these work files.

JOB CONTROL STATEMENT EXAMPLE

Figure 9 is an example of job control statement input (SYSRDR=SYSIPT) required to perform a series of job steps in an installation using labeled magnetic tape. In the discussion that follows, each point corresponds to the number at the left of the two slashes in the job control statements.

1. JOB statement for the series of job steps to be performed.

2. ASSGN statements that are required for the job steps. It is assumed that the assignments differ from those specified when the system was generated. The new assignments will be carried through for the entire job and will be reassigned at the end of the job to the standards established at system generation time.

3. OPTION statement specifying that the output of the assembly will be written on SYS000 for subsequent linkage editing. In addition, a listing of the source module will be output on SYSLST, and the dump option will be exercised in the event of an abnormal end of job. The DECK option will be overridden by the LINK option.

4. PHASE and EXEC statements for an assembly, followed by the Assembler source deck and the end-of-data-file indicator (/*), followed by an ENTRY statement.

5. Number of magnetic tape label blocks (NMTLB) statement followed by the EXEC statement for the Linkage Editor. The NMTLB statement is required to direct the Linkage Editor to reserve a portion of the lower problem program area as a label block area. The Linkage Editor edits the object program on SYS000 and writes the edited program back on SYS000.

6. Volume (VOL) and tape label (TPLAB) statements for the object program to be executed, followed by the EXEC statement for the linkage-edited object program on SYS000.

7. PAUSE statement that requests operator action.

8. OPTION statement specifying that the no-dump option be exercised. The LINK option is included to enable a new linkage-edit.

9. INCLUDE statements for modules in the relocatable library that are to be included with the object module on SYSIPT. A blank operand indicates that the module to be included follows on SYSIPT. The resulting program is edited and written on SYS000.

10. EXEC statement for the program to be executed. The data for the execution is followed by the end-of-data-file indicator.

11. Volume (VOL) and tape label (TPLAB) statements for the object program in the core image library to be executed, followed by the EXEC statement. The NMTLB statement is not required when the program to be executed is cataloged as a permanent entry in the core image library. (The NMTLB statement is used by the Linkage Editor and immediately precedes the EXEC statement for the Linkage Editor.)

12. PAUSE statement that requests operator action.

13. End-of-job indicator. All symbolic unit assignments are reset to the standards established when the system was generated.

14. JOB statement for the next job.

```
1   //      JOB       EXAMPLE
2   //      ASSGN     SYS004,X'101'
    //      ASSGN     SYS005,X'10F'
3   //      OPTION    LINK,DECK,LIST,DUMP
            PHASE     PHS1,ROOT
4   //      EXEC      ASSEMBLY
    (Assembler Source Deck)
    /*
            ENTRY
5   //      NMTLB     2
    //      EXEC      LNKEDT
    //      VOL       SYS004,SOURCE
    //      TPLAB     'label-information'
6   //      VOL       SYS005,RESULTS
    //      TPLAB     'label-information'
    //      EXEC
7   //      PAUSE     SAVE SYS004, SYS005, MOUNT NEW TAPES
8   //      OPTION    NODUMP, LINK
            PHASE     PHNAM
            INCLUDE   SQRT
            INCLUDE   SINE
9           INCLUDE
    (Object Deck to be Included)
    /*
            ENTRY
    //      EXEC      LNKEDT
    //      EXEC
10  (Data for User Object Program)
    /*
    //      VOL       SYS004,SOURCE
    //      TPLAB     'label-information'
11  //      VOL       SYS005,RESULTS
    //      TPLAB     'label-information'
    //      EXEC      PROGA
12  //      PAUSE     SAVE SYS004, SYS005, MOUNT NEW TAPES
13  /&
14  //      JOB       NEXT
```

Figure 9. Control Statement Example

Operation of the IBM Basic Operating System/360 is initiated through an initial program load (IPL) procedure from the resident tape. The operator places the resident tape on a drive, selects the address of that drive in the load unit switches, and presses the load key. This causes the first record to be read into main storage bytes 0-23. The information read in consists of an IPL PSW and two CCW's, which in turn cause the reading and loading of the IPL Loader.

Operating in the supervisor state, the IPL Loader reads the Supervisor nucleus into low main storage. If a read error is sensed while reading the Supervisor nucleus, the wait state is entered and an error code is set in the first word of main storage. The IPL procedure must then be restarted.

After successfully reading in the Supervisor nucleus, the IPL Loader performs these operations.

1. Sets the LUB table entry for SYSRES to point to the PUB entry of the channel and unit number of the resident drive.

2. Places the processing unit in the wait state with all interruptions masked except the I/O interruption. When the wait state is entered, the operator decides whether a 1052 or a card reader will be used to communicate with the system. If a 1052, the attention key is pressed; if a card reader, it is brought to the ready state.

3. At this time, the operator has several options. He may change the PUB configuration by adding or deleting a device. When a device is deleted, all references to it are removed. A device may be added only if additional space has been made available in the PUB table. This is specified as a system generation parameter. If a tape is to be added, there must also be enough space for an associated Tape Error Block (TEB).

To add a device to the PUB table, a control statement, read by the communication device (1052 or card reader), in the following format is required.

Operation	Operand
ADD	X'cuu'[(k)],devicetype[,X'ss']

where:

X'cuu' = channel and unit numbers.

k = S if the device is to be switchable (the device is physically attached to two adjacent channels). The designated channel is the lower of the two channels. If the device is not switchable, k = 0-255, indicating the priority of the device, with 0 indicating the highest priority.

devicetype = actual device (2400, 1443, etc).

X'ss' = device specifications (see ASSGN Statement). Used for 7-track tapes. If absent, X'00' is assigned.

To delete a device from the PUB table, a control statement, read by the communication device (1052 or card reader), in the following format is required.

Operation	Operand
DEL	X'cuu'

where cuu is the channel and unit numbers of the device to be deleted.

4. The only communication required at IPL time is the date and, if the timer is present, the time of day. It must follow any ADD or DEL statements. It is entered via the communication device (1052 or card reader) and is in the following format.

Operation	Operand
SET	DATE=value1,CLOCK=value2

value1 Has one of the following formats.

mm/dd/yy
dd/mm/yy

mm specifies the month; dd specifies the day; yy specifies the year. The format to be used is the format that was selected when the system was generated.

value2 Has the following format.

hh/mm/ss

hh specifies hours; mm specifies minutes; ss specifies seconds.

After completing these operations, IPL issues the EOJ macro instruction. This macro instruction issues a FETCH for the Job Control program, which begins processing the control statements for the first job of the day. Control statements are present on the device assigned to SYSRDR.

If the operator wishes to change any symbolic unit assignments, ASSGN statements are entered via the communication device (1052 or card reader). The ASSGN statement is as described in I/O Commands.

All programs executed in the Basic Operating System/360 environment must be edited by the Linkage Editor. The Linkage Editor reads the relocatable output of the language translators and edits it into executable, non-relocatable programs on SYS000. Once a program has been edited, it can be executed from SYS000, or it can be cataloged as a permanent entry in the core image library. When a program has been cataloged in the core image library, the Linkage Editor is no longer required for that program. The program is run as a distinct job step and is loaded directly from the resident tape by the System Loader.

The extent of the editing function performed depends on the structure of the input program. The simplest case is that of a single-module program. The Linkage Editor has only to edit the program on SYS000, creating a single phase entry in the core image format. This corresponds to the first diagram in Figure 10.

In more complex situations, the operation may involve linking together and relocating multiple-control sections from separate assemblies to produce a number of separate phases on SYS000 (see the last diagram in Figure 10). The Linkage Editor resolves all linkages (symbolic references) between segments of the program and relocates the phases to load at specified main-storage locations.

To facilitate writing and testing large programs, assembled program sections cataloged in the relocatable library can be combined with other sections from SYSIPT (card or tape). It is advantageous to handle some kinds of subroutines in this way instead of as macros.

STAGES OF PROGRAM DEVELOPMENT

The term program could be confused with several things. The programmer codes sets of source statements that may be a complete program or part of a program. These source statements are then compiled or assembled into a relocatable machine-language program which, in turn, must be edited into an executable program, and may be combined with other programs. Consequently, it is convenient to refer to each stage of program development by a particular name.

A set of source statements that is processed by a language translator (Assembler, COBOL, FORTRAN, RPG, or PL/I) is referred to as a source module.

The output of a language translator is referred to as an object module. All object modules must be further processed by the Linkage Editor before they can be executed in the basic operating system. Note: Each element in the relocatable library is called a module. These relocatable modules each consist of a single object module.

The output of the Linkage Editor consists of one or more program phases in SYS000. A phase is in executable, non-relocatable, core image form. Each separate phase is loaded by the System Loader in response to a FETCH or LOAD macro.

STRUCTURE OF A PROGRAM

Source Modules: A source module is input to a language translator and consists of definitions for one or more control sections. When the source module is translated, the output (object module) consists of one or more defined control sections. Each control section is a block of code assigned to contiguous main-storage locations. The input for building a phase (a section of a program loaded as a single overlay) must consist of one or more complete control sections.

Object Module: An object module is the output of a complete language translator run. It consists of control dictionaries and text of one or more control sections. The control dictionaries contain the information necessary for the Linkage Editor to resolve cross references between different object modules. The text is the actual instructions and data fields of the object module. The program cards produced by the language translators (as distinct from the Linkage Editor control statements to be discussed later) have an identifier in columns 1-4 that indicates the content of the card. The following card types, except REP cards, are produced by the language translators:

Identification	Contents or Meaning
ESD	External Symbol Dictionary

TXT	Text
RLD	Relocation Dictionary Item
REP	Replacement Text Record
END	End of a Module

With the exception of REP cards, all these cards must be present in an object module in the indicated sequence. REP cards, when required, appear after the TXT cards and before the END card, and are produced by the programmer. The format of the REP card is described in a latter part of this section.

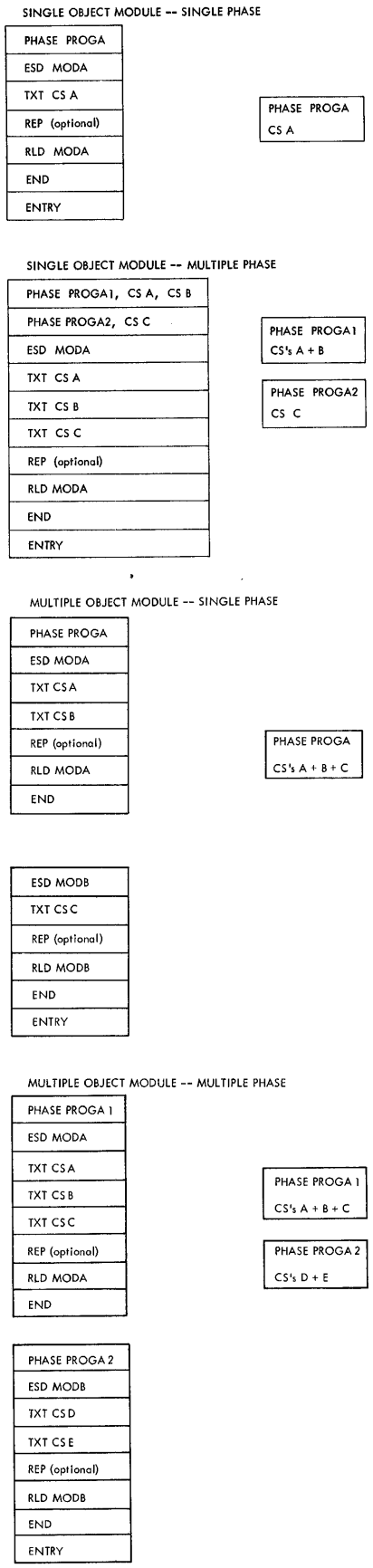


Figure 10. Linkage Editor Input and Output

Program Phase: A program phase, the output of the Linkage Editor, is that section of a program that is loaded as a single overlay with a single FETCH or LOAD by the System Loader. Programs may consist of many phases, the first fetched by Job Control, and each of the rest by a preceding program phase. Successive phases of a multi-phase program are often called overlays.

The input for building a single phase consists of the text from one or more complete control sections. When building a phase, the Linkage Editor constructs a composite ESD and a composite PHASE data, known as the control dictionary, and a composite RLD from each of the modules that make up the phase. These composite dictionaries are used to resolve all linkages between different control sections as if they had been assembled as one module. Each control section within the phase is relocated as necessary, and the entire phase is assigned a contiguous area of main storage. All relocatable address constants are modified to contain the relocated value of their symbols. The Linkage Editor always ensures that each phase or control section begins on a double-word boundary.

Each phase is constructed by building the text in a work area and then writing it in blocks in SYS000. Thus, a phase may consist of one or more blocks of contiguous core image locations. Backward origin within a phase causes slower operation of the Linkage Editor.

TYPES OF LINKAGE EDITOR RUNS

The Linkage Editor is run as a distinct job step. Because of this fact, it is meaningful to classify it as one of the system service programs. The Linkage Editor function is performed as a job step in three kinds of operations (Figure 11).

1. Catalog Programs in Core Image Library. The Linkage Editor function is performed immediately preceding the Librarian operation (MAINT) that catalogs (CATAL option) programs as permanent entries in the core image library. The sequence of events when this operation is performed is shown in the first diagram of Figure 11. Note that the input for the LNKEDT function could include modules from the relocatable library instead of, or in addition to, those

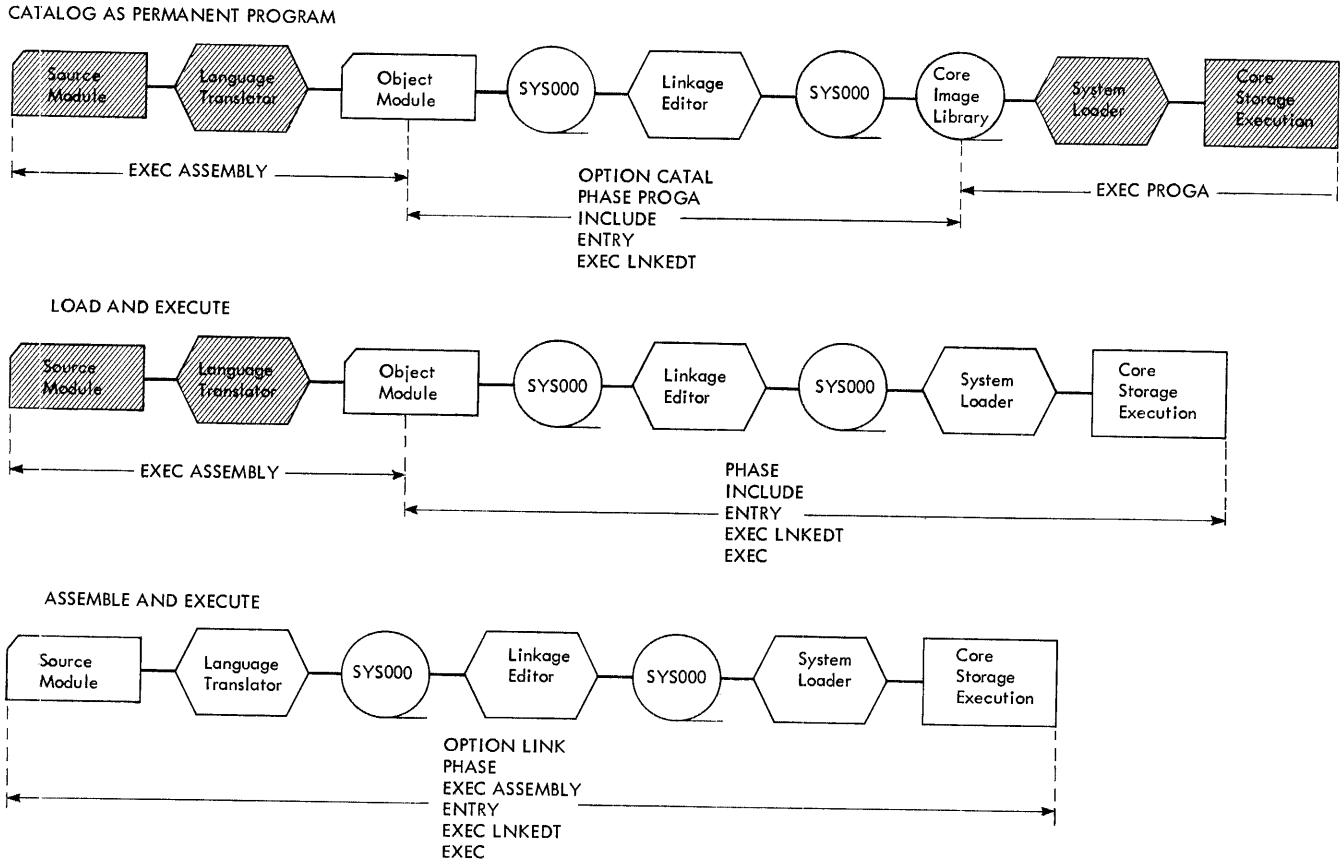


Figure 11. Three Types of Linkage Editor Operations

modules from the card reader or tape unit assigned to SYSIPT. This is accomplished by including the name of the module to be included in an INCLUDE statement. In addition, had the LINK option been specified in the OPTION control statement, the object module output by the language translator would have been written directly on SYS000.

2. Load-and-Execute. The sequence of events when this operation is performed is shown in the second diagram in Figure 11. Just as with the catalog operation, the input can consist of object modules from the relocatable library instead of, or in addition to, those modules from the card reader or tape unit assigned to SYSIPT. This is accomplished by including the name of the module to be included in the operand of an INCLUDE statement. After the object modules have been edited on SYS000, the program on SYS000 is executed. The blank operand in the EXEC control statement indicates that the program just linkage-edited on SYS000 is to be executed.

3. Assemble-and-Execute. Source modules can be assembled or compiled and then executed in a single sequence of job steps. In order to do this, the language translator is directed to output the object module directly on SYS000. This is done by using the LINK option in the OPTION control statement. Upon completion of this output operation, the Linkage Editor function is performed. The linkage-edited program is stored on SYS000. The sequence of events when this operation is performed is shown in the third diagram in Figure 11.

LINKAGE EDITOR CONTROL STATEMENTS

In addition to the program cards previously listed, object modules used as input for the Linkage Editor include Linkage Editor control statements. There are four kinds of these control statements.

PHASE — To indicate the beginning of a phase. It gives the name of the phase and the main-storage address where it is to be loaded.

INCLUDE — To signal that an object module is to be included. A blank operand indicates to Job Control that the module is on SYSIPT. An entry in the operand indicates that a module by that name is to be included from the relocatable library. INCLUDE statements with blank operands are recognized only on SYSRDR. Each series of relocatable modules on SYSIPT must be terminated by a /* control statement.

ENTRY — To signal the end of the last input object module and to provide an optional transfer address for the first phase.

ACTION — To produce a main-storage map on SYSLST.

The first (or only) object module input for the Linkage Editor must include a PHASE control statement before the first ESD item. The last (or only) object module may optionally be followed by an ENTRY control statement. The rules governing placement of INCLUDE and other PHASE control statements are discussed under Control Statement Placement.

SOURCES OF INPUT

Input for the Linkage Editor is always from the tape unit assigned to SYS000. Object module input to SYS000 can be:

1. Output from the language translator programs immediately after a compilation or assembly.
2. From only the card reader or tape unit assigned to SYSIPT.
3. From SYSIPT and from the relocatable library.
4. From only the relocatable library.

In the first case, the LINK option in the OPTION control statement indicates to Job Control and to language translators that the output resulting from an assembly or compilation will be written directly on SYS000.

In the second case, an INCLUDE statement with a blank operand indicates to Job Control that the object module on SYSIPT is to be written on SYS000.

In the third case, an INCLUDE statement with a blank operand indicates the presence of an object module on SYSIPT. An INCLUDE statement with an entry in the operand indicates that a module in the relocatable library is to be included in the program phase.

In the fourth case, an INCLUDE statement with an entry in the operand indicates that a module in the relocatable library is to be included in the program phase. Modules in the relocatable library can also contain INCLUDE statements.

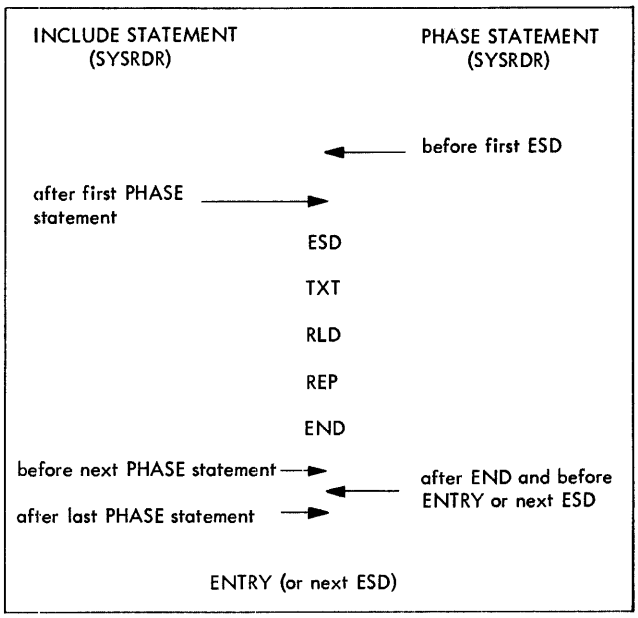
GENERAL CONTROL STATEMENT FORMAT

The Linkage Editor control statements are similar in format to statements processed by the Assembler. The operation field must be preceded by one or more blanks. The operation field must begin to the right of column 1 and must be separated from the operand field by at least one blank position. The operand field is terminated by the first blank position. It cannot extend past column 71.

CONTROL STATEMENT PLACEMENT

When preparing multiple-object modules in a single Linkage Editor run, the single ENTRY statement should follow the last object module on SYSIPT. The ACTION statement immediately precedes the first PHASE statement.

Figure 12 shows the possible placement of the PHASE and INCLUDE statements.



Note: INCLUDE statements within modules in the relocatable library must precede the ESD statement for the module.

Figure 12. Placement of PHASE and INCLUDE Statements

PHASE STATEMENT

The PHASE statement must precede the first object module or INCLUDE statement of each phase processed by the Linkage Editor. Under no circumstances can a PHASE statement occur within a control section. This is an error condition that can result in erroneous program loading. There can be several control sections within a phase.

This statement provides the Linkage Editor with a phase name and an origin point for the phase. The phase name is used to catalog the phase in the core image library. This name is used in a FETCH or LOAD macro to retrieve the phase for execution. The PHASE statement is in the following format.

Operation	Operand
PHASE	name,origin[,(namelist)][,NOAUTO]

The name field is blank. The operation field contains PHASE. Entries in the operand field must be separated by commas. The entries in the operand field represent the following.

name — Symbolic name of the phase. One to eight alphameric characters, the first of which must be alphabetic, are used as the phase name. If more than eight characters are used, those in excess of eight are truncated. Multi-phase programs must have phase names of five to eight alphameric characters, the first of which must be alphabetic. The first four characters of each phase within a multi-phase program must be the same. Otherwise, bytes 40-43 of the communication region will not contain an accurate indication of the uppermost byte used when loading problem program phases.

origin — Specifies the load address of the phase. The load address can be in one of three forms:

1. symbol[(phase)][±relocation]
2. *[±relocation]
3. ROOT
4. +displacement

The elements that make up the three forms that specify the origin signify the following.

1. symbol: may be a previously

defined phase name, a previously defined control section, or a previously defined external label (the operand of an entry statement).

(phase): if symbol is a previously defined control section or a previously defined external label that appears in more than one phase, phase, (in parentheses) directs the Linkage Editor to the phase in which the origin name is to be found. The phase name must have been defined previously.

relocation: indicates the origin of the phase currently being processed will be set relative to the symbol by a relocation term consisting of a + or a - immediately followed by a hexadecimal number X'hhhhhh' of one to six digits, or a decimal number ddddddd of one to eight digits.

2. *: indicates the Linkage Editor phase location counter. It will cause the Linkage Editor to assign the next main-storage location (with forced double-word alignment) as an origin for the next phase.

   relocation: indicates relocation of the phase as described above.

3. ROOT: causes the Linkage Editor to consider the phase that follows as a root phase that will always be resident in main storage while the program is being executed . This phase will not be overlayed by any subsequent phase. The main-storage address assigned to this phase is the first double-word address after the area assigned to the COMMON pool (if any) and the label block area (if any). Only the first PHASE statement is permitted to specify ROOT. If a control section appears in the root phase, other occurrences of the same control section are ignored and all references are resolved to the control section in the root. Control sections are not duplicated within the same phase.

4. +displacement: allows the origin point (loading address) to be set at a specified location. The origin point is an absolute address, relative to zero. displacement is a hexadecimal number X'hhhhhh' of one to six digits, or a decimal number ddddddd of one to eight digits.

(namelist) Causes the Linkage Editor to construct a phase from only the control sections specified. The namelist is in the following format.

(csnamel,csname2,...)

Entries within the parentheses are the names of the control sections that will be used to constitute a phase. When the namelist option is used and only selected control sections are included in a phase, a sub-modular phase is created. The counterpart of a sub-modular phase is a normal phase. A normal phase contains all control sections of one or more object modules. A program should be composed either completely of sub-modular phases or completely of normal phases. The total number of control sections referenced by a PHASE statement cannot exceed ten. Sub-modular phases cannot be constructed from any object module in the relocatable library. When several PHASE statements appear before an object module, each of the statements must contain the namelist option. Any object module not preceded by a PHASE statement will be included in the current phase.

NOAUTO Indicates that the Automatic Library Look-Up (AUTOLINK) is suppressed. The AUTOLINK feature provides the following. AUTOLINK collects each unresolved external reference from the phase. It then searches the relocatable library for an object module with the same name as each unresolved external reference. When a match is found, the module in the relocatable library is edited into the

phase. Object-module cross references with labels identical to library object-module entry-point labels are erroneous. The use of NOAUTO as the last operand in a PHASE statement causes the AUTOLINK process to be suppressed for that phase only.

Some examples of PHASE statements follow.

```
PHASE PHNAME,*+504
```

This causes loading to start 504 bytes past the end of the previous phase.

```
PHASE PHNAME3,PHNAME2
```

This causes loading to start in the same point where the loading of the phase by the name PHNAME2 started.

```
PHASE PHNAME ROOT
```

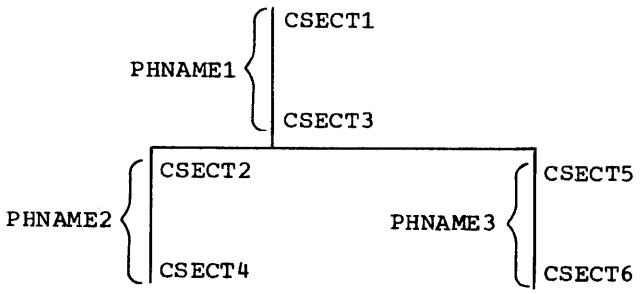
This causes loading to start after the label block area and the COMMON pool in the problem program area. This phase will never be overlayed by any other phase. When the PHASE statement contains a ROOT origin, this PHASE statement must be the first PHASE statement read by the Linkage Editor.

```
PHASE PHNAME,CSECT1(PHNAME2)
```

This causes loading to start at the point where CSECT1 was loaded. CSECT1, the named control section, must have appeared in the phase named PHNAME2.

```
PHASE PHNAME1,*,(CSECT1,CSECT3)
PHASE PHNAME2,*,(CSECT2,CSECT4)
PHASE PHNAME3,PHNAME2,(CSECT5,CSECT6)
```

This sequence of PHASE statements causes the Linkage Editor to structure the next module composed of CSECT1-CSECT6 in three overlays as shown below.



Because more than one PHASE statement appeared before the object module, each of the PHASE statements had to contain a control section namelist.

Note: In each of the preceding examples, if the origin address supplied is not a double-word boundary, the Linkage Editor will automatically increment to the next double-word boundary.

The Linkage Editor will allow the inclusion of the same control section within each of several phases except the root phase (if any). In this case, as external references occur in a phase, they are resolved preferentially with the first previous corresponding entrypoint within the phase. If this is not possible, they are resolved with the first corresponding entrypoint anywhere within the program, if such an entrypoint exits.

INCLUDE STATEMENT

This statement is used to indicate that an object module is to be included for editing by the Linkage Editor. The statement has a single, optional operand. When the operand is omitted, the object module to be included is assumed to be next on SYSIPT. If an operand is present, the object module is assumed to be in the relocatable library. It must be the same module name that was used when the module was cataloged in the relocatable library. Including modules from the relocatable library permits the programmer to include standard subroutines in his program at linkage-edit time.

The placement of the INCLUDE statement determines the position of the module in the program phase. An included module (in the relocatable library) can be preceded by one or more additional INCLUDE statements.

The format of the INCLUDE statement is:

Name	Operation	Operand
blank	INCLUDE	[modulename]

modulename — Symbolic name of the module, as used when cataloged in the relocatable library. It consists of one alphabetic character, followed by up to seven alphameric characters.

Modules in the relocatable library can be nested by using INCLUDE statements up to a level of five. Modules included by INCLUDE statements read from SYSRDR are referred to as being in the first level. Modules included by statements in the first level are at the second level. This is illustrated in Figure 13. Modules included by statements in the second level are at the third level, and so on up to five levels.

ENTRY STATEMENT

Every program, as input for the Linkage Editor, is terminated by an ENTRY statement. Its format is:

Name	Operation	Operand
blank	ENTRY	[entrypoint]

entrypoint — Symbolic name of an entry point. It must be the name of a CSECT, a label definition, or a label reference. If the operand field is blank, the Linkage Editor will use as a transfer address the first significant address provided in an END record encountered during the generation of the first phase. If no such operand is found, the transfer address will be the load address of the phase.

ACTION STATEMENT

This statement is used to indicate that SYSLST is available for output. When used, the statement must precede the first PHASE statement. Its format is:

Name	Operation	Operand
blank	ACTION	MAP

MAP — Indicates that SYSLST is available for diagnostic messages. In addition, a main storage map is output on SYSLST. The map contains every entry within each CSECT and every CSECT within each phase.

REP CARD

The REP card allows substitution (or replacement) of new text for portions of assembled text. Each REP card contains the assembled address of the first byte to be replaced, and may contain from 2 to 22 bytes of text. The text is substituted, byte for byte, for the original text, beginning at the address specified. Both the new address and the new text must be stated in hexadecimal, using the following format.

Card Columns	
1	Multiple punch (12-2-9). Identifies this as a loader card.
2-4	REP. Replace text card.
5-6	Blank.
7-12	Assembled address of the first byte to be replaced (hexadecimal).
13	Blank.
14-16	External symbol identification number (ESID) of the control section (SD) containing the text.
17-72	From 1 to 11 four-digit hexadecimal fields separated by commas, each replacing one previously loaded halfword. A blank indicates the end of information in this card.
73-80	May be used for program identification.

The REP card must be placed within the module that it modifies. If the module consists of input for more than one phase, REP must be placed within the bounds of the phase and after the text which it modifies. The Linkage Editor output as a result of the REP card is a newly created TXT card. This TXT card provides the necessary information and data to make the modification that was requested. (Address constants cannot be modified by REP cards since the RLD relocation applies only to TXT cards in the Linkage Editor input stream.)

Duplicate TXT card input containing address constants will not be relocated correctly. A REP or TXT card that causes loading of information outside the limits of a phase is processed. If the ACTION MAP control statement is used, a warning indication of the condition is made.

PHASE ENTRY POINT

The Linkage Editor stores with each phase the absolute entry point of that phase. These entry points are as follows.

1. For the first phase, the entry point specified in the ENTRY statement is used. If no entry point is specified, the first significant entry address taken from an END statement encountered in the construction of the first phase is used.

2. For sub-modular phases, the entry point specified in the END statement of the module for the first phase constructed from the module is used. If subsequent phases are constructed from the module, the beginning address of each phase is used.

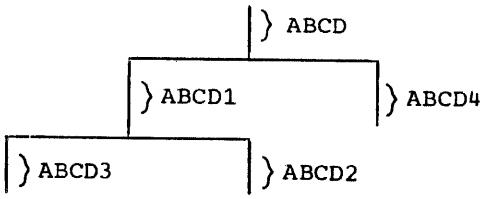
3. For all other phases, the entry address specified in the first END statement encountered in the construction of the phase is used.

SEQUENCE OF PHASES INPUT TO THE LINKAGE EDITOR

The input sequence of phases, by phase name, is important only if the phases are to be cataloged in the core image library. Phases to be cataloged must be submitted in ascending order by phase name. If the program has a root phase, it must be the first and, therefore, must have the "lowest" phase name. The following sequence of PHASE statements is acceptable:

```
PHASE ABCD,ROOT
PHASE ABCD1,*
PHASE ABCD2,*
PHASE ABCD3,ABCD2
PHASE ABCD4,ABCD1
```

This sequence of PHASE statements causes the Linkage Editor to structure the program as shown below.



The following are examples of incorrect phase sequences.

```
PHASE ABCD1,*
PHASE ABCD,ROOT
PHASE ABCD2,*
PHASE ABCD3,ABCD2
PHASE ABCD4,ABCD1
```

The preceding sequence is invalid because the PHASE statement containing ROOT must be the first PHASE statement in the sequence.

```
PHASE ABCD,ROOT
PHASE ABCD1,*
PHASE ABCD2,*
PHASE ABCD4,ABCD1
PHASE ABCD3,ABCD2
```

The preceding sequence is invalid because the phases by the names ABCD3 and ABCD4 are not in ascending order.

EXAMPLE OF LINKAGE EDITOR INPUT AND OUTPUT

The program shown in Figure 13 illustrates the rules governing input for the Linkage Editor and shows the output obtained. Though this example is somewhat more complex than the normal program, by following the flow of the input one can find practically every situation that may arise.

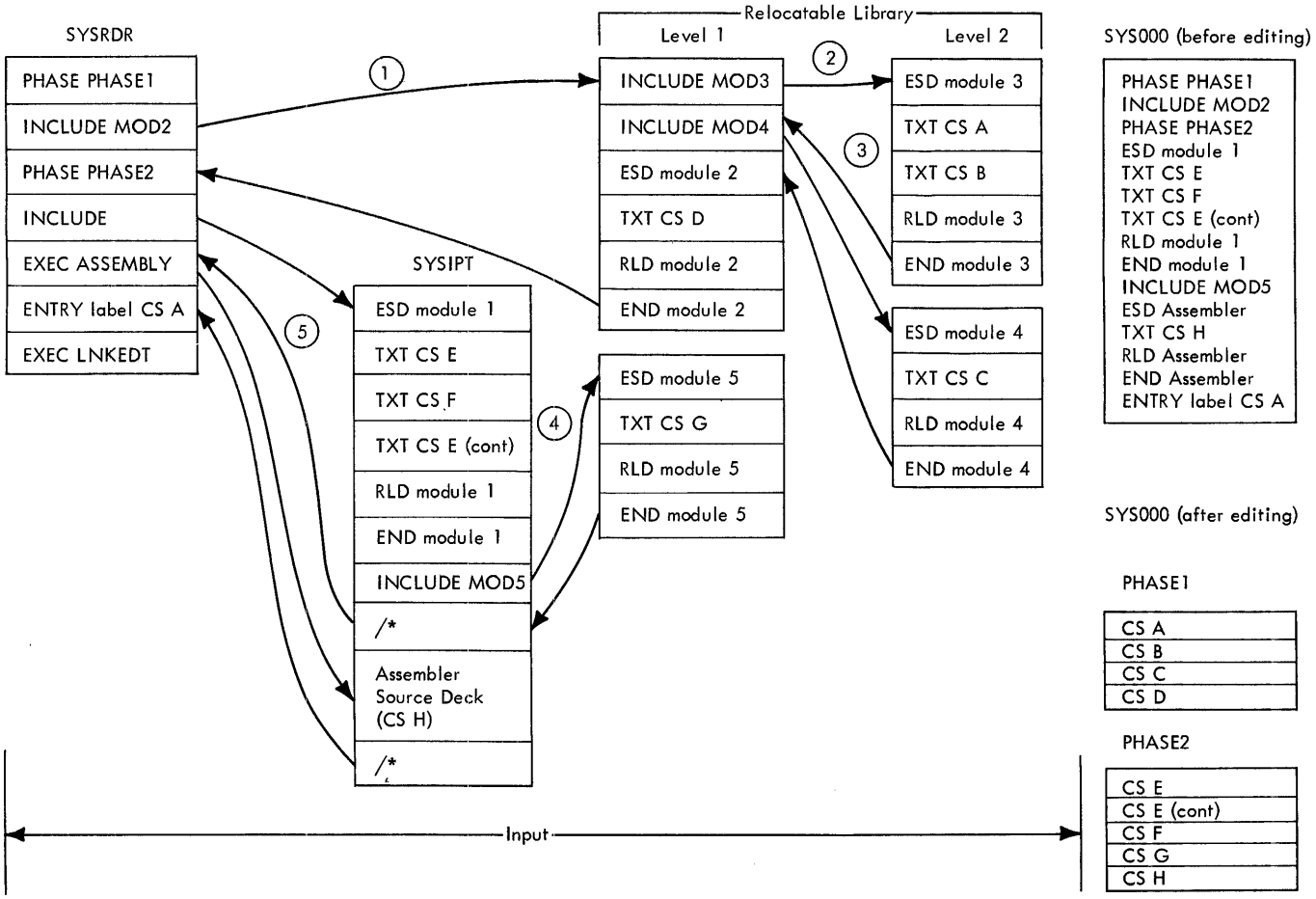


Figure 13. Example of Linkage Editor Input and Output

The leftmost block shows control statements being read from SYSRDR. The next block is read from SYSIPT and contains an object module (module 1) and a source module to be assembled. The next two blocks represent two levels in the relocatable library. The rightmost block shows the output phases as they appear on SYS000, both before and after the execution of the Linkage Editor function.

1. The first INCLUDE statement accesses the relocatable library. Therefore, the first entry in module 2 must be either on ESD or another INCLUDE. In this case, module 2 begins with an INCLUDE statement.

2. In this case, module 3 begins with an ESD.

3. INCLUDE statements can be grouped.

4. This split control section (CS E) is assigned a contiguous area of main storage.

5. A source module can be assembled and combined with object modules from SYSIPT and the relocatable library during the same job step. It is assumed that the LINK option in the OPTION statement has been specified before this job step. The LINK option directs the language translator to output the object program on SYS000.

This section describes the set of programs that maintain and service the libraries of a 16K tape-resident system. This set of programs is collectively referred to as the Librarian.

The system residence can contain three separate and distinct libraries:

1. Core image library
2. Source statement library
3. Relocatable library.

The core image library is required for each tape-resident system. The other two libraries, the source statement library and the relocatable library, are not required for operating a system.

CORE IMAGE LIBRARY

The core image library contains any number of programs. Each program is made up of one or more separate phases. Hence, each phase may either be a single program, or an overlay in the case of a multiphase program.

All programs in the core image library are edited to run with the resident Supervisor. Each program phase is assigned a fixed location in main storage. The programs in the core image library include system programs, the librarian programs, other IBM programs such as Assembler, RPG, COBOL, FORTRAN, PL/I, and sort programs, and user programs. All programs in the library are in alphameric order.

Associated with each phase in the core image library is a header describing the phase. Included in the header are the phase name, the starting main-storage address, the ending main-storage address, the maximum main-storage address used by any phase of the program, the entry-point address, the number of bytes per block, the number of blocks, and the length of the last block.

SOURCE STATEMENT LIBRARY

The source statement library contains any number of books. Each book in the source statement library is made up of a sequence of source language statements in 80-80 format.

The purpose of the source statement library is to provide an extension of the functions of a macro library. If a source program contains a macro instruction, the macro definition in the source statement library corresponding to the macro instruction is generated in the source program. If the source program contains a COPY statement, the specific language translator will compile a book from the source statement library into the source program.

Each book in the source statement library is classified as belonging to a specific sub-library. Sub-libraries are currently defined for two programming languages, Assembler and COBOL, in the system. Classifying books by a sub-library name allows books written in these languages to have the same name.

Card images are stored in compressed form within the library. In the compressed format, all blanks are eliminated. When a book is retrieved, the card images are expanded to their original 80-character format.

Associated with the source statement library are a source statement directory and a header record for each book in the library. The directory has entries defining each sub-library and the books associated with each sub-library. Included in each header record are the identification of the source statement library, the number of records from the last tape mark, the sub-library name, the book name, and the number of records in the book.

Books in a sub-library of the source statement library and entries in the source statement directory are in alphameric order.

It is possible for the source statement library to reside on a separate tape. When this is desired, the symbolic name used to define the source statement library (SYSSLB) is assigned to a device other than the device assigned to the system residence symbolic name (SYSRES).

RELOCATABLE LIBRARY

The relocatable library contains any number of modules. Each module is a complete object deck in the relocatable format.

The purpose of the relocatable library is to allow the user to maintain frequently used routines in residence and combine them with other modules without requiring recompilation. The routines from the relocatable library are edited on SYS000 by the Linkage Editor.

Associated with each module in the relocatable library is a header describing the module. Included in the header are the identification of the relocatable library, the number of records from the last tape mark, the module name, and the number of records in the module.

Modules in the relocatable library are in alphameric order.

It is possible for the relocatable library to reside on a separate tape. When this is desired, the symbolic name used to define the relocatable library (SYSRLB) is assigned to a device other than the device assigned to the system residence symbolic name (SYSRES).

LIBRARIAN FUNCTIONS

The Librarian programs perform two major functions:

1. Maintenance
2. Service.

Maintenance functions are used to add, delete, or copy components of the three libraries. The MAINT program is the maintenance program for all libraries defined for the system.

Service functions are used to translate information from a particular library to printed (displayed) or punched output. Information in a library directory and header records can also be displayed. The SSERV program is the service program for the source statement library. The RSERV program is the service program for the relocatable library. The DSERV program is the service program for the directories.

Librarian functions are performed through use of control statements. The control statements are:

1. A JOB control statement.
2. A number of ASSGN control statements that may be required to change the assignment of actual input/output devices.
3. An EXEC control statement requesting a particular librarian program.
4. Librarian specification statements describing various functions to be performed.
5. A /* control statement.
6. A /& control statement.

The JOB, ASSGN, /*, and /& control statements are the same as those described in the Job Control section. The operand field of the EXEC control statement is described in this section. The other statements pertain to the Librarian and are described in this section.

Librarian functions can be performed separately, or in certain combinations as described in the following sections. All control statement information is read from the device assigned (in the ASSGN statements) to SYSRDR. All input data is read from the device assigned to SYS000 or SYSIPT. SYSIPT and SYSRDR can be assigned to the same physical input/output device.

Figure 14 is a table of all maintenance functions. Figure 15 is a table of all service functions.

Function	Unit	Element	Control Statements Required

Catalog	Core Image Library	Program	// JOB jobname
			// OPTION CATAL
			// EXEC LNKEDT
			/&

	Source Statement Library	Book	// JOB jobname
			// EXEC MAINT
			CATALS sublib.bookname
			/*
			/&

	Relocatable Library	Module	// JOB jobname
			// EXEC MAINT
			CATALR modulename
			/*
			/&

Delete	Core Image Library	Phase	// JOB jobname
			// EXEC MAINT
			DELETC phase1[,phase2,...]
			/*
			/&

		Program	// JOB jobname
			// EXEC MAINT
			DELETC prog1.ALL[,prog2.ALL,...]
			/*
			/&

	Source Statement Library	Book	// JOB jobname
			// EXEC MAINT
			DELETS sublib.book1[,book2,...]
			/*
			/&

		Library	// JOB jobname
			// EXEC MAINT
			DELETS sublib.ALL
			/*
			/&

	Relocatable Library	Module	// JOB jobname
			// EXEC MAINT
			DELETR module1[,module2,...]
			/*
			/&

		Library	// JOB jobname
			// EXEC MAINT
			DELETR ALL
			/*
			/&

Copy	Core Image Library	Library	// JOB jobname
			// EXEC MAINT
			COPY CL
			/*
			/&

	Source Statement Library	Library	// JOB jobname
			// EXEC MAINT
			COPY SL
			/*
			/&

	Relocatable Library	Library	// JOB jobname
			// EXEC MAINT
			COPY RL
			/*
			/&

	Core Image Library and Source Statement Library	Libraries	// JOB jobname
			// EXEC MAINT
			COPY CL,SL
			/*
			/&

	Core Image Library and Relocatable Library	Libraries	// JOB jobname
			// EXEC MAINT
			COPY CL,RL
			/*
			/&

	System	Libraries	// JOB jobname
			// EXEC MAINT
			COPY CL,SL,RL
			/*
			/&

Figure 14. Maintenance Functions

Function	Unit	Element	Control Statements Required

Display	Core Image Library	Directory	// JOB jobname
			// EXEC DSERV
			DSPLY CD
			/*
			/&

	Source Statement Library	Book	// JOB jobname
			// EXEC SSERV
			DSPLY sublib.book1[,book2,...]
			/*
			/&

		Library	// JOB jobname
			// EXEC SSERV
			DSPLY sublib.ALL
			/*
			/&

		Directory	// JOB jobname
			// EXEC DSERV
			DSPLY SS
			/*
			/&

	Relocatable Library	Module	// JOB jobname
			// EXEC RSERV
			DSPLY module1[,module2,...]
			/*
			/&

		Library	// JOB jobname
			// EXEC RSERV
			DSPLY ALL
			/*
			/&

		Directory	// JOB jobname
			// EXEC DSERV
			DSPLY RD
			/*
			/&

	Directories	All	// JOB jobname
			// EXEC DSERV
			DSPLY ALL
			/*
			/&

Punch	Source Statement Library	Book	// JOB jobname
			// EXEC SSERV
			PUNCH sublib.book1[,book2,...][,COMPRSD]
			/*
			/&

		Library	// JOB jobname
			// EXEC SSERV
			PUNCH sublib.ALL[,COMPRSD]
			/*
			/&

	Relocatable Library	Module	// JOB jobname
			// EXEC RSERV
			PUNCH module1[,module2,...]
			/*
			/&

		Library	// JOB jobname
			// EXEC RSERV
			PUNCH ALL
			/*
			/&

Display and Punch	Source Statement Library	Book	// JOB jobname
			// EXEC SSERV
			DSPCH sublib.book1[,book2...][,COMPRSD]
			/*
			/&

		Library	// JOB jobname
			// EXEC SSERV
			DSPCH sublib.ALL[,COMPRSD]
			/*
			/&

	Relocatable Library	Module	// JOB jobname
			// EXEC RSERV
			DSPCH module1[,module2,...]
			/*
			/&

		Library	// JOB jobname
			// EXEC RSERV
			DSPCH ALL
			/*
			/&

Figure 15. Service Functions

MAINTENANCE FUNCTIONS

The set of maintenance functions contains three subsets:

1. Catalog
2. Delete
3. Copy.

The catalog function adds a phase to the core image library, adds a book to the source statement library, or adds a module to the relocatable library. If control statements and books or modules to be cataloged are read from the same device, the control statements must precede the associated book or module. Phases, books, or modules to be cataloged must be in alphameric sequence.

Programs to be cataloged in the core image library must first be edited by the Linkage Editor on SYS000. Input for the Linkage Editor can be from SYSIPT or from the relocatable library. See the section entitled Linkage Editor for a description of Linkage Editor functions that are performed prior to the catalog function for the core image library.

The delete function deletes a program, book, or module from a library. Programs, books, or modules to be deleted must be in alphameric sequence according to the particular library.

The copy function is used to copy the tape-resident system selectively or completely. A complete copy can be used to obtain backup if the original system is inadvertently destroyed. A selective copy can be used for reducing a complete system to a system that is designed to perform a specific purpose.

SERVICE FUNCTIONS

The set of service functions contains three subsets:

1. Display
2. Punch
3. Display and punch.

The tape-resident system can display and/or punch books in the source statement library, and modules in the relocatable library. In addition, the core image library header records, the source statement directory and header records, and the relocatable library header records can be displayed.

Whenever a requested service function provides punched-card output, the output is on the device assigned to SYSPCH. Whenever a requested service function provides printed output, the output is on the device assigned to SYSLST.

GENERAL CONTROL STATEMENT FORMAT

The librarian control statements are similar in format to statements processed by the Assembler. The operation field must be preceded by one or more blanks. The operation field must begin to the right of column 1 and must be separated from the operand field by at least one blank position. The operand field is terminated by the first blank position. It cannot extend past column 71. Continuation statements are not recognized.

LIBRARIAN FUNCTIONS: CORE IMAGE LIBRARY

This section describes the maintenance functions that relate to the core image library.

MAINTENANCE FUNCTIONS

To request a maintenance function, other than the catalog function, for the core image library, use the following EXEC control statement.

```
// EXEC MAINT
```

One or more of two maintenance functions (delete or copy) can be requested within a single run. Any number of programs within the core image library can be acted upon in this run. Further, one or more of the maintenance functions (catalog, delete, or copy) for either of the other two libraries (source statement or relocatable) can be requested within this run; the same MAINT program maintains all three of the libraries.

Catalog

The catalog function adds a phase to the core image library. While the MAINT program is the librarian program that catalogs a phase to the core image library, the manner of calling the MAINT program differs for the core image library. The CATAL option in the OPTION control statement indicates that upon encountering either // EXEC MAINT or /&, all programs which have been processed by the Linkage Editor will be cataloged in the core image library.

Each phase that is cataloged in the core image library derives its name from the PHASE control statement.

Input for the catalog function is from the tape unit assigned to SYS000. This tape contains output from the Linkage Editor.

If a phase in the core image library is to be replaced by a new phase having the same name, only the catalog function need be used. The delete function is implied with each catalog function.

Any number of phases can be cataloged in the core image library within a single run.

For the catalog function for the core image library, SYSRDR must be assigned to a card reader or a tape unit. SYS000 must be assigned to a tape unit. SYS002, the device on which the newly updated system library is located, must be assigned to a tape unit. SYSLST must be assigned to a printer or a tape unit, and SYSLOG can be assigned to a printer-keyboard.

Control statement input for the catalog function, read from the device assigned to SYSRDR, is as follows.

1. The JOB control statement, followed by
2. The ASSGN control statements, if the current assignments are not those required. The ASSGN statements that can be used are SYSRDR, SYS000, SYS002, SYSLST, and SYSLOG. The ASSGN statements are followed by
3. The OPTION control statement, with CATAL specified in the operand field, followed by
4. The appropriate Linkage Editor control statements (PHASE, INCLUDE, and ENTRY), followed by
5. The EXEC LNKEDT control statement, followed by
6. The /& control statement, which is the last control statement of the job.

Delete

The delete function is used to remove specific phases or programs from the core image library. Any number of phases or programs can be deleted during a single run.

The DELETC control statement in one of the following formats is used to delete programs or phases from the core image library.

```
DELETC phasename1[,phasename2,...]

DELETC prog1.ALL[,prog2.ALL,...]
```

In the first format, the entry in the operation field is DELETC. phasename in the operand field represents the name(s) of the phase(s) to be deleted. The name of the phase can be of any length; however, a maximum of the first eight characters is used to locate and delete the phase. Entries in the operand field must be separated by commas, and must be in alphameric sequence by phase name.

In the second format, prog refers to the first four characters of the program name. (All phases within a program have the same first four characters. Therefore, the first four characters of each program within the library should be unique.) The four characters are followed by a period and ALL.

Any number of DELETC control statements can be used for the core image library within a single run.

For the delete function, SYSRDR must be assigned to a card reader or a tape unit. SYS002, the device on which the newly updated system library is located, must be assigned to a tape unit. SYSLST must be assigned to a printer or a tape unit, and SYSLOG can be assigned to the printer-keyboard.

Control statement input for the delete function, read from the device assigned to SYSRDR, is as follows.

1. The JOB control statement, followed by
2. The ASSGN control statement, if the current assignments are not those required. The ASSGN statements that can be used are SYSRDR, SYS002, SYSLST, and SYSLOG. The ASSGN statements are followed by
3. The EXEC MAINT control statement, followed by
4. The DELETC control statement(s), followed by
5. The /* control statement, followed by
6. The /& control statement, which is the last control statement of the job.

Copy

The copy function is used to copy the system residence either selectively or completely. The tape unit on which the system is to be copied is assigned to SYS002.

A complete copy consists of copying each library and directory on the resident tape. The selective copy consists of copying only particular libraries.

The COPY control statement in one of the following formats is used for the copy function.

```
COPY CL

COPY CL,SL

COPY CL,RL

COPY CL,SL,RL
```

The first format is used when only the core image library is to be copied. The entry in the operation field is COPY. The entry in the operand field is CL.

The second format is used when the core image library and the source statement library are to be copied. The entry in the operation field is COPY. The entries in the the operand field, separated by a comma, are CL and SL.

The third format is used when the core image library and the relocatable library are to be copied. The entry in the operation field is COPY. The entries in the operand field, separated by a comma, are CL and RL.

The fourth format is used when the entire system is to be copied. The entry in the operation field is COPY. The entries in the operand field, separated by commas, are CL, SL, and RL.

For the copy function, SYSRDR must be assigned to a card reader or a tape unit. SYS002 must be assigned to a tape unit. SYSLST must be assigned to a printer or a tape unit, and SYSLOG can be assigned to a printer-keyboard.

Control statement input for the copy function, read from the device assigned to SYSRDR, is as follows.

1. The JOB control statement, followed by
2. The ASSGN control statements, if the current assignments are not those required. The ASSGN statements that can be used are SYSRDR, SYS002, SYSLST, and SYSLOG. The ASSGN statements are followed by
3. The EXEC MAINT control statement, followed by
4. The COPY control statement, followed by
5. The /* control statement, followed by
6. The /& control statement, which is the last control statement of the job.

LIBRARIAN FUNCTIONS: SOURCE STATEMENT LIBRARY

This section describes the maintenance and service functions that relate to the source statement library.

MAINTENANCE FUNCTIONS

To request a maintenance function for the source statement library, use the following EXEC control statement.

```
// EXEC MAINT
```

One or more of the three maintenance functions for the source statement library can be requested within a single run. Any number of books within the source statement library can be acted upon in this run. Further, one or more of the maintenance functions for either of the other two libraries (core image or relocatable) can be requested within this run; the same MAINT program maintains all three libraries.

Catalog

The catalog function adds a book to a sub-library of the source statement library. Card input for the catalog function is from the device assigned to SYSIPT. Books to be cataloged in the source statement library must be in alphameric order. Any number of books can be added within a single run.

The CATALS control statement is required to add a book to a sub-library of the source statement library. It is read from the device assigned to SYSRDR and is in the following format.

```
CATALS sublib.bookname
```

The operation field contains CATALS. The qualifier sublib in the operand field represents the sub-library to which the book is to be cataloged and can be:

A for the Assembler sub-library

C for the COBOL sub-library.

bookname in the operand field represents the name of the book to be cataloged. The name of the book can be of any length; however, a maximum of the first eight characters is used to cataloged the book.

Books that are to be cataloged in a sub-library of the source statement library must be preceded and followed by special statements indicating the beginning and the end of a book.

Macro definitions that are to be cataloged in the Assembler sub-library are preceded by the MACRO statement and are followed by the MEND statement. MACRO is the standard macro definition header statement; MEND is the standard macro definition trailer statement.

Books other than macro definitions that are to be cataloged in either the Assembler or COBOL sub-library of the source statement library are preceded and followed by a BKEND statement. A BKEND statement must precede each book, and a BKEND statement must follow each book. If desired, the BKEND statement may precede and follow a macro definition (in addition to the MACRO and MEND statements). This would be desirable when the options provided in the BKEND statement are required. The statement is in the following format.

```
BKEND [sub.book][,SEQNCE][,count][,COMPRSD]
```

The first column of the statement must be blank. The entry in the operation field is BKEND. All operand entries are optional. When used, the entries must be in the prescribed order. The first entry in the operand field, sub.book, is identical to the operand of the CATALS control statement. If the second operand, SEQNCE, is specified, columns 76 to 80 of the card images making up the book are checked for ascending sequence numbers. The count operand specifies the number of card images in the book. When used, the card input is counted, beginning with the preceding BKEND statement and including the following BKEND

statement. If an error is detected in either the sequence checking or the card count, an error message is printed. The error can be corrected, and the book can be recataloged. The COMPRSD operand indicates that the book to be cataloged in the library is in the compressed format.

It is possible to have the source statement library on a tape that does not contain the core image library. When this is true, the source statement library on the private tape is referenced by the symbolic name SYSSLB. The private library is built in one of two ways. First, a COPY function can be performed, copying the source statement library on the private tape. The alternate method is to define a completely new private tape. When this is the case, a special control statement is required that defines a new volume (NEWVOL) that contains the sub-libraries of the source statement library. This control statement is in the following format and precedes the first book to be cataloged to the first sub-library.

NEWVOL

The operation field contains NEWVOL. The operand field is blank.

For the catalog function, SYSRDR must be assigned to a card reader or a tape unit. SYSIPT must be assigned to a card reader or a tape unit. SYSSLB must be assigned to a tape unit, if the source statement library is on a private tape. SYS000 and SYS001 are used as work files and must each be assigned to a tape unit. SYS002, the device on which the newly updated library is located, must be assigned to a tape unit. SYSLST must be assigned to a printer or a tape unit, and SYSLOG can be assigned to a printer-keyboard.

Control statement input, read from the device assigned to SYSRDR, is as follows.

1. The JOB control statement, followed by
2. The ASSGN control statements, if the current assignments are not those required. The ASSGN statements that can be used are SYSRDR, SYSIPT, SYSSLB, SYS000, SYS001, SYS002, SYSLST, and SYSLOG. The ASSGN statements are followed by
3. The EXEC MAINT control statement, followed by
4. The NEWVOL control statement, if a new library is being defined, followed by
5. The CATALS control statement(s), followed by
6. The /* control statement, followed by
7. The /& control statement, which is the last control statement of the job.

Card image input, read from the device assigned to SYSIPT, is as follows.

1. The BKEND statement, and/or the MACRO header statement if the book is a macro definition, followed by
2. The book to be cataloged, followed by
3. The BKEND statement, and/or the MEND trailer statement if the book is a macro definition.

If SYSRDR and SYSIPT are assigned to the same device, the books to be cataloged immediately follow their respective CATALS control statements.

Delete

The delete function is used to remove specific books from a sub-library of the source statement library. The function can also be used to delete an entire sub-library. Any number of books from a specific sub-library can be deleted during a single run.

The DELETS control statement in one of the following formats is used to delete books from the source statement library.

```
DELETS sublib.book1[,book2,...]

DELETS sublib.ALL
```

The first format is used if only specific books are to be deleted from a specific sub-library. The entry in the operation field is DELETS. The qualifier sublib in the operand field represents the sub-library containing the book to be deleted and can be:

A for the Assembler sub-library

C for the COBOL sub-library.

book in the operand field represents the name of the book in the sub-library to be deleted. If more than one book from a specific sub-library is to be deleted, subsequent book names need not be qualified. Books to be deleted must be in alphameric sequence. Entries in the operand field must be separated by commas. The name of the book can be of any length; however, a maximum of the first eight characters is used to locate and delete the book. Continuation statements are not recognized.

The second format is used if an entire sub-library is to be deleted. The entry in the operation field is DELETS. The first entry in the operand field is the name of the sub-library to be deleted. sublib represents the sub-library to be deleted and can be:

A for the Assembler sub-library

C for the COBOL sub-library.

The second entry in the operand field is ALL. The two entries must be separated by a period.

For the delete function, SYSRDR must be assigned to a card reader or a tape unit. SYSSLB must be assigned to a tape unit, if the source statement library is on a private tape. SYS001 is used as a work file and must be assigned to a tape unit. SYS002, the device on which the newly updated library is located, must be assigned to a tape unit. SYSLST must be assigned to a printer or a tape unit, and SYSLOG can be assigned to the printer-keyboard.

Control statement input for the delete function, read from the device assigned to SYSRDR, is as follows.

1. The JOB control statement, followed by
2. The ASSGN control statements, if the current assignments are not those required. The ASSGN statements that can be used are SYSRDR, SYSSLB, SYS001, SYS002, SYSLST, and SYSLOG. The ASSGN statements are followed by
3. The EXEC MAINT control statement, followed by
4. The DELETS control statement(s), followed by
5. The /* control statement, followed by
6. The /& control statement, which is the last control statement of the job.

Copy

The copy function is used to copy the source statement library on another tape. The tape unit on which the library is to be copied is assigned to SYS002.

The COPY control statement in the following format is used for the copy function.

```
COPY SL
```

The entry in the operation field is COPY. The entry in the operand field is SL.

For the copy function, SYSRDR must be assigned to a card reader or a tape unit. SYSSLB must be assigned to a tape unit, if the source statement library is on a private tape. SYS002 must be assigned to a tape unit. SYSLST must be assigned to a printer or a tape unit, and SYSLOG can be assigned to a printer-keyboard.

Control statement input for the copy function, read from the device assigned to SYSRDR, is as follows.

1. The JOB control statement, followed by
2. The ASSGN control statements, if the current assignments are not those required. The ASSGN statements that can be used are SYSRDR, SYSSLB, SYS002, SYSLST, and SYSLOG. The ASSGN statements are followed by
3. The EXEC MAINT control statement, followed by
4. The COPY control statement, followed by
5. The /* control statement, followed by
6. The /& control statement, which is the last control statement of the job.

SERVICE FUNCTIONS

To request a service function for the source statement library, use the following EXEC control statement.

```
// EXEC SSERV
```

One or more of the three service functions can be requested within a single run. Any number of books within the source statement library can be acted upon in this run.

Display

The display function is used to get a print-out of a book in the source statement library. Any number of books can be displayed within a single run. The printed output consists of a header and the information contained within the book.

Contained within the printed header is the name of the book and the number of records within the book.

Books are displayed in the card image format. Each book is preceded and followed by a BKEND statement.

The DSPLY control statement in one of the following formats is used to display books in the source statement library.

```
DSPLY sublib.book1[,book2,...]

DSPLY sublib.ALL
```

The first format is used if only specific books are to be displayed from a specific sub-library. The entry in the operation field is DSPLY. The qualifier sublib in the operand field represents the sub-library containing the book to be displayed and can be:

A for the Assembler sub-library

C for the COBOL sub-library.

book in the operand field represents the name of the book in the sub-library to be displayed. If more than one book from a specific sub-library is to be displayed, subsequent book names need not be qualified. Books to be displayed must be in alphameric sequence. Entries in the operand field must be separated by commas. The name of the book can be of any length; however, a maximum of the first eight characters is used to locate and display the book. Continuation statements are not recognized.

The second format is used if an entire sub-library is to be displayed. The entry in the operation field is DSPLY. The first entry in the operand field is the name of the sub-library to be displayed. sublib can be:

A for the Assembler sub-library

C for the COBOL sub-library.

The second entry in the operand field is ALL. The two entries must be separated by a period.

For the display function, SYSRDR must be assigned to a card reader or a tape unit. SYSSLB must be assigned to a tape unit, if the source statement library is on a private tape. SYSLST must be assigned to a printer or a tape unit. SYSLOG can be assigned to a printer-keyboard.

Control statement input for the display function, read from the device assigned to SYSRDR, is as follows.

1. The JOB control statement, followed by
2. The ASSGN control statements, if the current assignments are not those required. The ASSGN statements that can be used are SYSRDR, SYSLST, SYSSLB, and SYSLOG. The ASSGN statements are followed by
3. The EXEC SSERV control statement, followed by
4. The DSPLY control statement(s), followed by
5. The /* control statement, followed by
6. The /& control statement, which is the last control statement of the job.

Punch

The punch function is used to convert a book in the source statement library into a punched-card output deck. The resulting punched-card deck consists of card images of the book in the library. If the optional operand, COMPRSD, is specified, the card images are punched in the compressed form in which they are stored in the library. Each book is preceded and followed by a BKEND card.

Any number of books in the source statement library can be punched within a single run.

The PUNCH control statement in one of the following formats is used to punch books in the source statement library.

```
PUNCH sublib.book1[,book2,...][,COMPRSD]

PUNCH sublib.ALL[,COMPRSD]
```

The first format is used if only specific books are to be punched from a specific sub-library. The entry in the operation field is PUNCH. The qualifier sublib in the operand field represents the sub-library containing the book to be punched and can be:

A for the Assembler sub-library

C for the COBOL sub-library.

book in the operand field represents the name of the book in the sub-library to be punched. If more than one book from a specific sub-library is to be punched, subsequent book-names need not be qualified. The entry COMPRSD is used if the books are to be punched in the compressed form in which they are stored in the library. When this option is selected, the cards are punched in the first seventy-one columns. Books to be punched must be in alphameric

sequence. Entries in the operand field must be separated by commas. The name of the book can be of any length; however, a maximum of the first eight characters is used to locate and punch the book. Continuation statements are not recognized.

The second format is used if an entire sub-library is to be punched. The entry in the operation field is PUNCH. The first entry in the operand field is the name of the sub-library to be punched. sublib can be:

A for the Assembler sub-library

C for the COBOL sub-library.

The second entry in the operand field is ALL. The entry COMPRSD is used if the books are to be punched in the compressed format.

For the punch function, SYSRDR must be assigned to a card reader or a tape unit. SYSSLB must be assigned to a tape unit, if the source statement library is on a private tape. SYSPCH must be assigned to a card punch or a tape unit. SYSLST must be assigned to a printer or a tape unit, and SYSLOG can be assigned to a printer-keyboard.

Control statement input for the punch function, read from the device assigned to SYSRDR, is as follows.

1. The JOB control statement, followed by
2. The ASSGN control statements, if the current assignments are not those required. The ASSGN statements that can be used are SYSRDR, SYSSLB, SYSPCH, SYSLST, and SYSLOG. The ASSGN statements are followed by
3. The EXEC SSERV control statement, followed by
4. The PUNCH control statement(s), followed by
5. The /* control statement, followed by
6. The /& control statement, which is the last control statement of the job.

Display and Punch

The display-and-punch function is used to combine the separate operations of the display function and the punch function. The output of the display-and-punch function is identical to that described in the two preceding subsections. Any number of books in a specific sub-library of the source statement library can be displayed and punched within a single run.

The DSPCH control statement in one of the following formats is used to convert books in a sub-library of the source statement library to printed and punched-card output.

```
DSPCH sublib.book1[,book2,...][,COMPRSD]

DSPCH sublib.ALL[,COMPRSD]
```

The first format is used if only specific books are to be displayed and punched from a specific sub-library. The entry in the operation field is DSPCH. The qualifier sublib in the operand field represents the sub-library containing the book to be displayed and punched and can be:

A for the Assembler sub-library

C for the COBOL sub-library.

book in the operand field represents the name of the book in the sub-library to be displayed and punched. If more than one book from a specific sub-library is to be displayed and punched, subsequent booknames need not be qualified. The entry COMPRSD is used if the books are to be punched in the compressed form in which they are stored in the library, but printed in the original card image format. Books to be displayed and punched must be in alphameric sequence. Entries in the operand field must be separated by commas. The name of the book can be of any length; however, a maximum of the first eight characters is used to locate and display and punch the book. Continuation statements are not recognized.

The second format is used if an entire sub-library is to be displayed and punched. The entry in the operation field is DSPCH. The first entry in the operand field is the name of the sub-library to be displayed and punched. sublib can be:

A for the Assembler sub-library

C for the COBOL sub-library.

The second entry in the operand field is ALL. The entry COMPRSD is used if the books are to be punched in the compressed format.

For the display and punch function, SYSRDR must be assigned to a card reader or a tape unit. SYSSLB must be assigned to a tape unit, if the source statement library is on a private tape. SYSLST must be assigned to a printer or a tape unit. SYSPCH must be assigned to a card punch or

a tape unit. SYSLOG can be assigned to a printer-keyboard.

Control statement input for the display and punch function, read from the device assigned to SYSRDR, is as follows.

1. The JOB control statement, followed by
2. The ASSGN statements, if the current assignments are not those required. The ASSGN statements that can be used are SYSRDR, SYSSLB, SYSLST, SYSPCH, and SYSLOG. The ASSGN statements are followed by
3. The EXEC SSERV control statement, followed by
4. The DSPCH control statement(s), followed by
5. The /* control statement, followed by
6. The /& control statement, which is the last control statement of the job.

LIBRARIAN FUNCTIONS: RELOCATABLE LIBRARY

This section describes the maintenance and service functions that relate to the relocatable library.

MAINTENANCE FUNCTIONS

To request a maintenance function for the relocatable library, use the following EXEC control statement.

```
// EXEC MAINT
```

One or more of the three maintenance functions for the relocatable library can be requested within a single run. Any number of modules within the relocatable library can be acted upon in this run. Further, one or more of the maintenance functions for either of the other two libraries (core image or source statement) can be requested within this run, for the same MAINT program maintains all three libraries.

Catalog

The catalog function adds a module to the relocatable library. Input for the catalog function is from the device assigned to SYSIPT. A module in the relocatable library is the output of a complete language translator run.

A module added to the relocatable library is removed by using the delete function.

The catalog function implies a delete function. Thus, if a module exists in the relocatable library with the same name as a module to be cataloged, the module in the relocatable library is deleted.

The CATALR control statement is required to add a module to the relocatable library. The CATALR control statement is read from the device assigned to SYSRDR and is in the following format.

```
CATALR modulename
```

The operation field contains CATALR. The entry in the operand field, modulename, is the name by which the module will be known to the control system. The modulename is one to eight alphameric characters, the first of which must be alphabetic.

The cards composing the input for a module are described in the section entitled Linkage Editor. The cards are:

1. INCLUDE control statement (if appropriate)
2. ESD card
3. TXT card
4. REP card
5. Rep card
6. END card.

These input cards are read from the device assigned to SYSIPT.

It is possible to have the relocatable library on a tape that does not contain the core image library. When this is true, the relocatable library on the private tape is referenced by the symbolic name SYSRLB. The private library is built in one of two ways. First, a COPY function can be performed, copying the relocatable library on the private tape. The alternate method is to define a completely new private tape. When this is the case, a special control statement is required that defines a new volume (NEWVOL) that contains the relocatable library. This control statement is in the following format and precedes the first module to be cataloged to the newly defined relocatable library.

```
NEWVOL
```

The operation field contains NEWVOL. The operand field is blank.

For the catalog function, SYSRDR must be assigned to a card reader or a tape unit.

SYSIPT must be assigned to a card reader or a tape unit. SYSRLB must be assigned to a tape unit, if the relocatable library is on a private tape. SYS000 and SYS001 are used as work files and must each be assigned to a tape unit. SYS002, the device on which the newly updated library is located, must be assigned to a tape unit. SYSLST must be assigned to a printer or a tape unit, and SYSLOG can be assigned to a printer-keyboard.

Control statement input, read from the device assigned to SYSRDR, is as follows.

1. The JOB control statement, followed by
2. The ASSGN control statements, if the current assignments are not those required. The ASSGN statements that can be used are SYSRDR, SYSIPT, SYSRLB, SYS000, SYS001, SYS002, SYSLST, and SYSLOG. The ASSGN statements are followed by
3. The EXEC MAINT control statement, followed by
4. The NEWVOL control statement, if a new library is being defined, followed by
5. The CATALR control statement(s), followed by
6. The /* control statement, followed by
7. The /& control statement, which is the last control statement of the job.

Delete

The delete function is used to remove specific modules from the relocatable library. The function can also be used to delete the entire relocatable library. Any number of modules can be deleted during a single run.

The DELETR control statement in one of the following formats is used to delete modules from the relocatable library.

```
DELETR modname[,modname,...]

DELETR ALL
```

The first format is used when a specific module is to be deleted. The entry in the operation field is DELETR. The entry in the operand field, modname, is the name of the module to be deleted. If more than one module is to be deleted, the module names are separated by a comma. Modules to be deleted must be in alphameric sequence. modname is one to eight alphameric characters, the first of which must be alphabetic Continuation statements are not recognized.

The second format is used if the entire library is to be deleted. The entry in the operation field is DELETR. The entry in the operand field is ALL.

Any number of DELETR control statements can be used for the relocatable library within a single run.

For the delete function, SYSRDR must be assigned to a card reader or a tape unit. SYSRLB must be assigned to a tape unit, if the relocatable library is on a private tape. SYS001 is used as a work file and must be assigned to a tape unit. SYS002, the device on which the newly updated library is located, must be assigned to a tape unit. SYSLST must be assigned to a printer or a tape unit, and SYSLOG can be assigned to the printer-keyboard.

Control statement input for the delete function, read from the device assigned to SYSRDR, is as follows.

1. The JOB control statement, followed by
2. The ASSGN control statements, if the current assignments are not those required. The ASSGN statements that can be used are SYSRDR, SYSRLB, SYS001, SYS002, SYSLST, and SYSLOG. The ASSGN statements are followed by
3. The EXEC MAINT control statement, followed by
4. The DELETR control statement(s), followed by
5. The /* control statement, followed by
6. The /& control statement, which is the last control statement of the job.

Copy

The copy function is used to copy the relocatable library on another tape. The tape unit on which the library is to be copied is assigned to SYS002.

The COPY control statement in the following format is used for the copy function.

```
COPY RL
```

The entry in the operation field is COPY. The entry in the operand field is RL.

For the copy function, SYSRDR must be assigned to a card reader or a tape unit. SYSRLB must be assigned to a tape unit, if the relocatable library is on a private tape. SYS002 must be assigned to a tape unit. SYSLST must be assigned to a printer or a tape unit, and SYSLOG can be assigned to a printer-keyboard.

Control statement input for the copy function, read from the device assigned to SYSRDR, is as follows.

1. The JOB control statement, followed by
2. The ASSGN control statements, if the current assignments are not those required. The ASSGN statements that can be used are SYSRDR, SYSRLB, SYS002, SYSLST, and SYSLOG. The ASSGN statements are followed by
3. The EXEC MAINT control statement, followed by
4. The COPY control statement, followed by
5. The /* control statement, followed by
6. The /& control statement, which is the last control statement of the job.

SERVICE FUNCTIONS

To request a service function for the relocatable library, use the following EXEC control statement.

```
// EXEC RSERV
```

One or more of the three service functions can be requested within a single run. Any number of modules within the relocatable library can be acted upon in this run.

Display

The display function is used to get a print-out of a module in the relocatable library. Any number of modules can be displayed within a single run. The printed output consists of a header and the module.

Contained in the printed header is the module name and the number of records needed to contain the module.

The printed output of the module is represented by hexadecimal characters and EBCDIC, depending on the type of record and the information contained within the record.

The DSPLY control statement in one of the following formats is used to display modules in the relocatable library.

```
DSPLY module1[,module2,...]

DSPLY ALL
```

The first format is used if only specific modules are to be displayed. The entry in the operation field is DSPLY. The entry in the operand field, module, represents the name of the module to be displayed. If more than one module is to be displayed, the module names are separated by commas. Modules to be displayed must be in alphameric sequence. Module names are from one to eight characters long. Continuation statements are not recognized.

The second format is used if the entire relocatable library is to be displayed. The entry in the operation field is DSPLY The entry in the operand field is ALL.

For the display function, SYSRDR must be assigned to a card reader or a tape unit. SYSRLB must be assigned to a tape unit, if the relocatable library is on a private tape. SYSLST must be assigned to a printer or a tape unit. SYSLOG can be assigned to a printer-keyboard.

Control statement input for the display function, read from the device assigned to SYSRDR, is as follows.

1. The JOB control statement, followed by
2. The ASSGN control statements, if the current assignments are not those required. The ASSGN statements that can be used are SYSRDR, SYSRLB, SYSLST, and SYSLOG. The ASSGN statements are followed by
3. The EXEC RSERV control statement, followed by
4. The DSPLY control statement(s), followed by
5. The /* control statement, followed by
6. The /& control statement, which is the last control statement of the job.

Punch

The punch function is used to convert a module in the relocatable library into a punched-card output deck.

Any number of modules in the relocatable library can be punched within a single run.

The punched-card output is acceptable to every function that uses relocatable modules as input.

The PUNCH control statement in one of the following formats is used to convert modules in the relocatable library to punched-card output.

```
PUNCH module1[,module2,...]

PUNCH ALL
```

The first format is used if only specific modules are to be punched. The entry in the operation field is PUNCH. The entry in the operand field, module, represents the name of the module to be punched. If more than one module is to be punched, the module names are separated by commas. Modules to be punched must be in alphameric sequence. Module names are from one to eight characters long. Continuation statements are not recognized.

The second format is used if the entire relocatable library is to be punched. The entry in the operation field is PUNCH. The entry in the operand field is ALL.

For the punch function, SYSRDR must be assigned to a card reader or a tape unit. SYSRLB must be assigned to a tape unit, if the relocatable library is on a private tape. SYSPCH must be assigned to a card punch or a tape unit. SYSLST must be assigned to a printer or a tape unit, and SYSLOG can be assigned to a printer- keyboard.

Control statement input for the punch function, read from the device assigned to SYSRDR, is as follows.

1. The JOB control statement, followed by
2. The ASSGN control statements, if the current assignments are not those required. The ASSGN statements that can be used are SYSRDR, SYSRLB, SYSPCH, SYSLST, and SYSLOG. The ASSGN statements are followed by
3. The EXEC RSERV control statement, followed by
4. The PUNCH control statement(s), followed by
5. The /* control statement, followed by
6. The /& control statement, which is the last control statement of the job.

Display and Punch

The display-and-punch function is used to combine the separate operations of the display function and the punch function. The output of the display-and-punch function is identical to that described in the two preceding subsections. Any number of modules in the relocatable library may be displayed and punched within a single run.

The DSPCH control statement is used to convert modules in the relocatable library to printed and punched-card output. The DSPCH control statement is in one of the following formats.

```
DSPCH module1[,module2,...]

DSPCH ALL
```

The first format is used if only specific modules are to be displayed and punched. The entry in the operation field is DSPCH. The entry in the operand field, module, represents the name of the module to be displayed and punched. If more than one module is to be displayed and punched, the module names are separated by commas. Modules to be displayed and punched must be in alphameric sequence. Module names are from one to eight characters long. Continuation statements are not recognized.

The second format is used if the entire relocatable library is to be displayed and punched. The entry in the operation field is DSPCH. The entry in the operand field is ALL.

For the display-and-punch function, SYSRDR must be assigned to a card reader or a tape unit. SYSRLB must be assigned to a tape unit, if the relocatable library is on a private tape. SYSLST must be assigned to a printer or a tape unit. SYSPCH must be assigned to a card punch or a tape unit. SYSLOG can be assigned to a printer-keyboard.

Control statement input for the display and punch function, read from the device assigned to SYSRDR, is as follows.

1. The JOB control statement, followed by
2. The ASSGN control statements, if the current assignments are not those required. The ASSGN statements that can be used are SYSRDR, SYSRLB, SYSLST, SYSPCH, and SYSLOG. The ASSGN statements are followed by
3. The EXEC RSERV control statement, followed by

4. The DSPCH control statement(s), followed by

5. The /* control statement, followed by

6. The /& control statement, which is the last control statement of the job.

LIBRARIAN FUNCTIONS: DIRECTORIES

This section describes the service function, display, that relates to the three directories. A directory on the tape resident system is made up of the header records that precede each program, book, or module in the core image library, the source statement library, or the relocatable library, respectively.

To request the display service function for a directory, use the following EXEC control statement.

```
// EXEC DSERV
```

Any number of directories can be displayed within a single run.

Each printed line contains one header record. The phase, book, or module name is displayed in EBCDIC. All other fields are represented by hexadecimal characters.

The DSPLY control statement in the following format is used to display specific directories or all directories.

```
DSPLY dir1[,dir2[,dir3]]

DSPLY ALL
```

The first format is used if only specific directories are to be displayed. The entry in the operation field is DSPLY. The entry in the operand field, dir, represents the name of the directory to be displayed. It can be:

CD for the core image directory
SS for the source statement directory
RD for the relocatable directory.

If more than one directory is to be displayed, the symbols for the directories must be separated by a comma and can be in any order.

The second format is used if all three directories are to be displayed. The entry in the operation field is DSPLY. The entry in the operand field is ALL.

For the display function, SYSRDR must be assigned to a card reader or a tape unit. SYSLST must be assigned to a printer or a tape unit. If the operand ALL or SS is used, and if the source statement library is on a private tape, SYSSLB must be assigned to a tape unit. If the operand ALL or RD is used, and if the relocatable library is on a private tape, SYSRLB must be assigned to a tape unit. SYSLOG can be assigned to a printer-keyboard.

Control statement input for the display function, read from the device assigned to SYSRDR, is as follows.

1. The JOB control statement, followed by

2. The ASSGN control statements, if the current assignments are not those required. The ASSGN statements that can be used are SYSRDR, SYSLST, SYSSLB, SYSRLB, and SYSLOG. The ASSGN statements are followed by

3. The EXEC DSERV control statement, followed by

4. The DSPLY control statement, followed by

5. The /* control statement, followed by

6. The /& control statement, which is the last control statement of the job.

SYSTEM GENERATION

This section describes the general procedure that are followed in generating a tape resident basic operating system.

The tape resident system is received on a tape. The tape received contains three defined libraries: the core image, relocatable, and source statement. The libraries contain the following components.

1. Core image library. Contains programs in the core image format. The programs in the core image library are the Supervisor and its associated transient routines (OPEN, CLOSE, etc), Job Control, Linkage Editor, MAINT, and Assembler. All programs in the core image library are edited to run with the Supervisor supplied by IBM.

2. Relocatable library. Contains modules in the relocatable format. The modules in the relocatable library are the resident control program and service programs. These programs include Job Control, Linkage Editor, and the librarian programs. The librarian programs are MAINT, SSERV, RSERV, and DSERV. Other programs that can be in the relocatable library are Assembler,

COBOL, FORTRAN, PL/I, RPG, Autotest, Sort/Merge, and Utilities. The programs in the relocatable library will subsequently be edited to run with a Supervisor that is adapted to the configuration of the individual installation.

3. Source statement library. Contains books in source-language format. The books supplied are macro definitions in the Assembler sub-library. Among the macro definitions in the Assembler sub-library are the Supervisor macros and the logical IOCS macros.

The tape received by the user is capable of operating as a system. However, it is recommended that a Supervisor be assembled that is adapted to the configuration of the individual installation, and that the libraries of the system be edited according to the needs of the installation. When this process is completed, the newly created system replaces the system that was received by the user.

The recommended procedure for generating a system that meets the requirements of the individual installation follows.

1. Create a set of variable symbol statements. Since a Supervisor is assembled by using conditional assembly, a group of SET variable symbols is created to select various options, such as storage protection, timer feature, date convention, etc.

2. Catalog the set of variable symbol statements (called CONFIG) in the Assembler sub-library of the source statement library. The MAINT program is used for this catalog function.

3. Assemble a new Supervisor. Assembly of the Supervisor depends conditionally upon the SET variable symbols defined by CONFIG. Therefore, a COPY CONFIG statement is included in each component of the Supervisor.

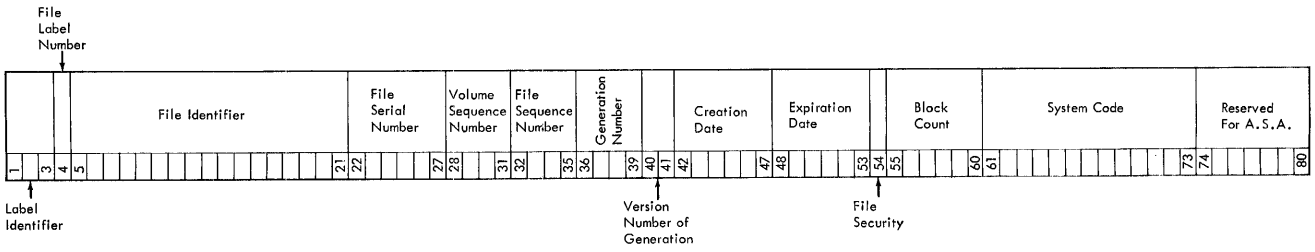
4. Linkage-edit the Supervisor and catalog it in the core image library. Other system programs are linkage-edited and cataloged to run with the new Supervisor. Only those programs required for the system need be cataloged; the selection depends upon the needs of the individual installation. The MAINT program is used for this catalog function.

5. Obtain backup by copying or punching the system. The MAINT or SSERV and RSERV programs are used for this function.

6. Delete the unused elements of the system for which backup has been obtained. The MAINT program is used for this function.

7. Catalog installation programs, macro definitions, etc, by using the maintenance (MAINT) program.

APPENDIX A: STANDARD TAPE FILE LABEL



The standard tape file label format and contents are as follows:

FIELD	NAME AND LENGTH	DESCRIPTION
1.	LABEL IDENTIFIER 3 bytes, EBCDIC	identifies the type of label HDR = Header -- beginning of a data file EOF = End of File -- end of a set of data EOV = End of Volume -- end of the physical reel
2.	FILE LABEL NUMBER 1 byte, EBCDIC	Always a 1
3.	FILE IDENTIFIER 17 bytes, EBCDIC	uniquely identifies the entire file, may contain only printable characters.
4.	FILE SERIAL NUMBER 6 bytes, EBCDIC	uniquely identifies a file/volume relationship. This field is identical to the Volume Serial Number in the volume label of the first or only volume of a multi-volume file or a multi-file set. This field will normally be numeric (000001 to 999999) but may contain any six alphameric characters.
5.	VOLUME SEQUENCE NUMBER 4 bytes	indicates the order of a volume in a given file or multi-file set. The first must be numbered 0001 and subsequent numbers must be in proper numeric sequence.
6.	FILE SEQUENCE NUMBER 4 bytes	assigns numeric sequence to a file within a multi-file set. The first must be numbered 0001.
7.	GENERATION NUMBER 4 bytes	uniquely identifies the various editions of the file. May be from 0001 to 9999 in proper numeric sequence.
8.	VERSION NUMBER OF GENERATION 2 bytes	indicates the version of a generation of a file.
9.	CREATION DATE 6 bytes	indicates the year and the day of the year that the file was created: Position / Code / Meaning 1 / blank / none 2-3 / 00-99 / Year 4-6 / 001-366 / Day of Year (e.g., January 31, 1965 would be entered as 65031)
10.	EXPIRATION DATE 6 bytes	indicates the year and the day of the year when the file may become a scratch tape. The format of this field is identical to Field 9. On a multifile reel, processed sequentially, all files are considered to expire on the same day.
11.	FILE SECURITY 1 byte	indicates security status of the file. 0 = no security protection 1 = security protection. Additional identification of the file is required before it can be processed.
12.	BLOCK COUNT 6 bytes	indicates the number of data blocks written on the file from the last header label to the first trailer label exclusive of tape marks. Count does not include checkpoint records. This field is used in Trailer Labels.
13.	SYSTEM CODE 13 bytes	uniquely identifies the programming system.
14.	RESERVED 7 bytes	Reserved for American Standards Association (A.S.A.). At present, should be recorded as blanks.

APPENDIX B: OPERATOR-TO-SYSTEM COMMANDS

<table>
<tr><th>Name</th><th>Operation</th><th>Operand</th><th>Remarks</th></tr>
<tr><td></td><td>ASSGN</td><td>SYSxxx,address [,X'ss'] [,ALT] [,TEMP]</td><td>SYSxxx: can be SYSRDR
SYSIPT
SYSPCH
SYSLST
SYSLOG
SYSSLB
SYSRLB
SYS000-SYS244

address: can be X'cuu', UA, or IGN

X'cuu': c = 0-6
uu = 00-FE (0-254) in hex

UA: unassign

IGN: unassign and ignore

X'ss': used for 7-track tape only

<table>
<tr><th>ss</th><th>Bytes per Inch</th><th>Parity</th><th>Trans-late Feature</th><th>Convert Feature</th></tr>
<tr><td>10</td><td>200</td><td>odd</td><td>off</td><td>on</td></tr>
<tr><td>20</td><td>200</td><td>even</td><td>off</td><td>off</td></tr>
<tr><td>28</td><td>200</td><td>even</td><td>on</td><td>off</td></tr>
<tr><td>30</td><td>200</td><td>odd</td><td>off</td><td>off</td></tr>
<tr><td>38</td><td>200</td><td>odd</td><td>on</td><td>off</td></tr>
<tr><td>50</td><td>556</td><td>odd</td><td>off</td><td>on</td></tr>
<tr><td>60</td><td>556</td><td>even</td><td>off</td><td>off</td></tr>
<tr><td>68</td><td>556</td><td>even</td><td>on</td><td>off</td></tr>
<tr><td>70</td><td>556</td><td>odd</td><td>off</td><td>off</td></tr>
<tr><td>78</td><td>556</td><td>odd</td><td>on</td><td>off</td></tr>
<tr><td>90</td><td>800</td><td>odd</td><td>off</td><td>on</td></tr>
<tr><td>A0</td><td>800</td><td>even</td><td>off</td><td>off</td></tr>
<tr><td>A8</td><td>800</td><td>even</td><td>on</td><td>off</td></tr>
<tr><td>B0</td><td>800</td><td>odd</td><td>off</td><td>off</td></tr>
<tr><td>B8</td><td>800</td><td>odd</td><td>on</td><td>off</td></tr>
</table>

ALT: specifies alternate unit

TEMP: assignment for logical unit will be destroyed by next JOB statement</td></tr>
<tr><td></td><td>RESET</td><td>blank</td><td>Resets I/O device assignments</td></tr>
<tr><td></td><td>CLOSE</td><td>SYSxxx</td><td>SYSxxx: SYSLST, SYSPCH, or SYS000-SYS244</td></tr>
<tr><td></td><td>DVCDN</td><td>X'cuu'</td><td>X'cuu': c = 0-6
uu = 00-FE (0-254) in hex</td></tr>
<tr><td></td><td>DVCUP</td><td>X'cuu'</td><td>X'cuu': c = 0-6
uu = 00-FE (0-254) in hex</td></tr>
<tr><td></td><td>MTC</td><td>opcode,X'cuu'</td><td>opcode: BSF, BSR, ERG, FSF, FSR, RUN, REW, or WTM

X'cuu': c = 0-6
uu = 00-FE (0-254) in hex</td></tr>
<tr><td></td><td>CANCEL</td><td>blank</td><td>Cancels execution of current job</td></tr>
<tr><td></td><td>PAUSE</td><td>comments</td><td>Causes pause at end of current job step</td></tr>
</table>

Name	Operation	Operand	Remarks
	LISTIO	{SYS / PROG / UA / DOWN / SYSxxx}	Causes listing of I/O assignments
	LOG	blank	Causes logging of job control statements on SYSLOG
	NOLOG	blank	Suppresses logging of job control statements on SYSLOG
	SET	[DATE=value1] [,CLOCK=value2] [,UPSI=value3] [,LINECT=value4]	value1: in one of the following formats mm/dd/yy dd/mm/yy mm: month (01-12) dd: day (01-31) yy: year (00-99) value2: in the following format hh/mm/ss hh: hours (00-23) mm: minutes (00-59) ss: seconds (00-59) value3: 0, 1, or X value4: standard number of lines for output on each page of SYSLST
	Ⓑ	blank	Ⓑ is alter code 5

APPENDIX C: JOB CONTROL STATEMENTS

Name	Operation	Operand	72	Remarks
//	JOB	jobname	⌀	jobname: one to eight alphameric characters
//	EXEC	[progname]	⌀	progname: one to eight alphameric characters. Used only if the program is in the core image library.
//	ASSGN	SYSxxx,address[,X'ss'] [,ALT]	⌀	SYSxxx: can be SYSRDR SYSIPT SYSPCH SYSLST SYSLOG SYSSLB SYSRLB SYS000-SYS244 address: can be X'cuu', UA, or IGN X'cuu': c = 0-6 uu = 00-FE (0-254) in hex UA: unassign IGN: unassign and ignore X'ss': used for 7-track tape only ss / Bytes per Inch / Parity / Translate Feature / Convert Feature 10 / 200 / odd / off / on 20 / 200 / even / off / off 28 / 200 / even / on / off 30 / 200 / odd / off / off 38 / 200 / odd / on / off 50 / 556 / odd / off / on 60 / 556 / even / off / off 68 / 556 / even / on / off 70 / 556 / odd / off / off 78 / 556 / odd / on / off 90 / 800 / odd / off / on A0 / 800 / even / off / off A8 / 800 / even / on / off B0 / 800 / odd / off / off B8 / 800 / odd / on / off ALT: specifies alternate unit
//	RESET	blank	⌀	Resets I/O device assignments
//	DATE	mm/dd/yy or dd/mm/yy	⌀ ⌀	mm: month (01-12) dd: day (01-31) yy: year (00-99)
//	UPSI	nnnnnnnn	⌀	n: 0, 1, or X
//	VOL	SYSxxx,filename	⌀	SYSxxx: can be SYSIPT SYSPCH SYS000-SYS244 filename: one to eight alphabetic characters
//	TPLAB	'label fields 3-10'	⌀	'label fields 3-10': indicated fields of the standard tape file label. A 49-byte character string, contained within single quotes.
//	TPLAB	'label fields 3-10 label fields 11-13'	C	'label fields 3-10: same as above C: any non-blank character in column 72 label fields 11-13': 20-character direct continuation of the same character string begun with fields 3-10 (no blanks, quotes, or commas separating)

Name	Operation	Operand	72	Remarks
//	RSTRT	jobname,SYSxxx,nnnn	⌀	jobname: one to eight alphameric character name of the job to be restarted SYSxxx: symbolic unit name of the device on which the checkpoint records are stored. Can be SYS000-SYS244. nnnn: one to four character identification of the checkpoint record to be used for restarting
//	LISTIO	{ SYS PROG UA DOWN SYSxxx }	⌀	Causes listing of I/O assignments
//	OPTION	option1 [,option2,...]	⌀	option: can be any of the following LOG — Log control statements on SYSLST NOLOG — Suppress LOG option DUMP — Dump registers and main storage on SYSLST in the case of abnormal program end NODUMP — Suppress DUMP option LINK — Write output of language translator on SYS000 for linkage editing NOLINK — Suppress LINK option DECK — Output object module on SYSPCH NODECK — Suppress DECK option LIST — Output listing of source module on SYSLST NOLIST — Suppress LIST option LISTX — Output listing of object module on SYSLST NOLISTX — Suppress LISTX option SYM — Print symbol table on SYSLST and/or punch symbol deck on SYSPCH NOSYM — Suppress SYM option XREF — Output symbolic cross-reference list on SYSLST NOXREF — Suppress XREF option ERRS — Output listing of all errors in source program on SYSLST NOERRS — Suppress ERRS option CATAL — Catalog program or phase in core image library immediately after completion of Linkage Editor run 48C — 48-character set 60C — 60-character set
//	PAUSE	[comments]	⌀	PAUSE statement is always printed on 1052 (SYSLOG). If no 1052 is available, the statement is ignored.
/*	ignored	ignored	⌀	Columns 1 and 2 are the only columns checked.
/&	ignored	ignored	⌀	Columns 1 and 2 are the only columns checked.
*		comments	⌀	Column 2 must be blank.

GLOSSARY

The explanations of the following terms relate to their use in the IBM Basic Operating System/360 (26K Tape) publications. These explanations may differ from those in other publications.

Basic Operating System/360 (16K Tape): A tape-resident system that provides operating system capabilities for 16K and larger System/360 tape configurations.

Block: 1. To group records physically for the purpose of conserving storage space or increasing the efficiency of access or processing.

2. A physical record on tape.

Book: A source statement routine written in the Assembler or COBOL language.

Buffer: 1. A storage device in which data is assembled temporarily during data transfers. It is used to compensate for a difference in the rate of flow of information or the time occurrence of events when transferring information from one device to another. For example, the IBM 2821 Control Unit (a control and buffer storage unit for card readers, card punches, and printers in a System/360).

2. A portion of main storage used for an input or output area.

Catalog: To enter a phase, module, or book into one of the system libraries as a permanent entry.

Channel Program: One or more Channel Command Words (CCW's) that control(s) a specific sequence of channel operations. Execution of the specific sequence is initiated by a single START I/O instruction.

Channel Scheduler: The part of the Supervisor that controls the movement of data between main storage and input/output devices.

Checkpoint: A point in a program at which sufficient information can be stored to permit restarting the problem program from that point.

Checkpoint Record: Tape records that contain the status of the job and the system at the time the records are written by the checkpoint routine. These records provide the necessary information for restarting a job without having to return to the beginning of the job.

Checkpoint/Restart: A means of restarting execution of a program at some point other than the beginning. When a checkpoint macro instruction is issued in a problem program, checkpoint records are created. These records contain the status of the program and the machine. When it is necessary to restart a program at a point other than the beginning, the restart procedure uses the checkpoint records to reinitialize the system.

Checkpoint Routine: A routine that records information for a checkpoint.

Command Control Block: A sixteen-byte field (five halfwords) required for each I/O device controlled by physical IOCS. This field is used for communication between physical IOCS and the problem program.

Communication Region: An area of the Supervisor set aside for interprogram and intraprogram communication. It contains information useful to both the Supervisor and the problem program.

Control Program: A group of programs that provides functions such as the handling of input/output operations, error detection and recovery, program loading, and communication between the program and the operator. IPL, Supervisor, and Job Control make up the control program in the Basic Operating System/360.

Control Section: The smallest separately relocatable unit of a program; that portion or text specified by the programmer to be entity, all elements fo which are to be loaded into contiguous main storage locations.

Core Image Library: An area of the resident system tape used to store programs that have been processed by the Linkage Editor. Each program is in a form identical to that which it must have to be executable in main storage. The programs in the core image library include system programs, the librarian programs, and other IBM-supplied programs such as Assembler, RPG, and sort

programs. User programs are also stored in the core image library.

Core Storage: See Main Storage

Data File: A collection of related data records organized in a specific manner. For example, a payroll file (one record for each employee, showing his rate of pay, deductions, etc.) or an inventory file (one record for each inventory item, showing the cost, selling price, number in stock, etc.).

External Reference: A reference to a symbol used in another module.

Fetch: 1. To bring a program phase into main storage from the core image library for immediate execution.

2. The routine that retrieves requested phases and loads them into main storage.

3. The name of a macro instruction (FETCH) used to transfer control to the System Loader.

File: See Data File

Fixed-Length Record: A record having the same length as all other records with which it is logically or physically associated.

Initial Program Loading (IPL): The initialization procedure that causes Basic Operating System/360 to read programs into main storage.

Input Job Stream: A sequence of job control statements entering the system, which may also include input data.

Input/Output Control System (IOCS): A group of macro-instruction routines provided by IBM for handling the transfer of data between main storage and external storage devices. IOCS consists of two parts: physical IOCS and logical IOCS.

Interruption: A break in the normal sequence of instruction execution. It causes an automatic transfer to a preset storage location where action is taken to satisfy the condition that caused the interruption.

I/O Area: An area (portion) of main storage into which data is read or from which data is written. In Operating System/360 publications, the term buffer is often used in place of I/O area. I/O means Input/Output.

IPL Loader: A program that reads the Supervisor into main storage and then transfers control to the Supervisor.

Job Control: A program that is called into storage to prepare each job or job step to be run. Some of its functions are to assign I/O devices to certain symbolic names, set switches for program use, log (or print) job control statements, and fetch the requested program phase.

Job Control Statement: Any one of the control statements in the input stream that identifies a job or job step or defines its requirements and options.

Job Statement (JOB): The control statement in the input stream that identifies the beginning of a series of job control statements for a single job i.e., a single accounting unit of processing.

Job Step: The execution of a single processing program.

K: 1024

Language Translators: A general term for any assembler, compiler, or other routine that accepts statements in one language and produces equivalent statements in another language. The Basic Operating System/360 has five language translators: Assembler, COBOL, FORTRAN, PL/I and RPG.

Librarian: The set of programs that maintains, services, and organizes the system libraries.

Library: An organized collection of programs, source statements, or object modules maintained on the system-resident tape. Three libraries are used by the Basic Operating System/360: core image library, source statement library, and relocatable library.

Linkage Editor: A system service program that edits the output of language translators and produces executable program phases. It relocates programs or program sections and links together separately assembled (or compiled) sections.

Load: To fetch, i.e., to read a phase into main storage preparatory to executing it.

Logic Module: An accessing routine that provides an interface between a processing program and physical IOCS.

Logical File: A data file that has been

described to the Basic Operating System/360 through the use of a file-definition (DTF) macro instruction. Note that a data file is described to Operating System/360 through a different defining method. Operating System/360 publications refer to a data file described in this different manner as a data set.

Logical IOCS: A comprehensive set of macro-instruction routines provided to handle the creation, retrieval, and maintenance of data files.

Logical Record: A record identified from the standpoint of its content, function, and use rather than its physical attributes. It is meaningful with respect to the information it contains. (Contrasted with Physical Record.)

Main Storage: All addressable storage from which instructions can be executed or from which data can be loaded directly into registers.

Module (Programming). The input to or output from a single execution of a language translator or the Linkage Editor; a separate program unit that can be combined with other units.

Object Module: The output of a single execution of a language translator; it constitutes input to the Linkage Editor. An object module consists of one or more control sections in relocatable, non-executable form and an associated control dictionary.

Operating System: A collection of programs that enables a data processing system to supervise its own day-to-day operations, automatically calling in programs, routines, languages, and data as needed for continuous throughput of an uninterrupted series of jobs.

Operating System/360: A modular and device independent operating system requiring direct access storage device residence; the minimum main storage requirement is 32K.

Overlap: To do something at the same time that something else is being done; for example, to perform input/output operations while instructions are being executed by the central processing unit.

Overlay: 1. A segment (phase) of a program retrieved into main storage, replacing all or part of a previously retrieved section.

2. The technique of repeatedly using the same blocks of internal storage during different stages of a problem. For example, when one routine is no longer needed in internal storage, another routine can replace all or part of that routine.

Overrun: To lose data by performing an I/O operation before a previous I/O operation on the same channel is complete.

Phase: The smallest complete unit that can be referenced in the core image library. Each overlay of a program or (if the program contains no overlays) the program itself is a single complete phase.

Physical IOCS: Macro instructions and Supervisor routines that schedule and supervise the execution of channel programs. Physical IOCS controls the actual transfer of records between the external storage medium and main storage.

Physical Record: A record identified from the standpoint of the manner or form in which it is stored and retrieved; that is, one that is meaningful with respect to access. (Contrasted with Logical Records.)

Problem Program: 1. The user's object program. It can be produced by any of the language translators. It consists of instructions necessary to solve the user's problem.

2. A general term for any routine that is executed in the data processing system's problem state; that is, any routine that does not contain privileged operations. (Contrasted with Supervisor.)

Processing Program: A general term for any program that is both loaded and supervised by the control program. Specifically, a collection of certain IBM-supplied programs: the language translators, Autotest, Sort/Merge, and Utilities. The term processing programs is in contrast to the term control program.

Queue: A list of entires, usually ordered in the sequence of arrival. The entries identify things contending for service or attention.

Record: A general term for any unit of data that is distinct from all others when considered in a particular context.

Relocatable: A module or control section whose address constants can be modi-

fied to compensate for a change in origin.

Relocatable-Library Module: A module consisting of one or more complete control sections cataloged as a single entry in the relocatable library.

Restart: See Checkpoint/Restart

Self-Relocating: A programmed routine that is loaded at any double-word boundary and can adjust its constants so as to be executed at that location.

Service Programs: Any of the class of standard routines that assist in the use of a data processing system and successful execution of problem programs. These programs include Autotest, Sort/Merge, and Utilities.

Source Module: A set of source statements in the symbolic language of a language translator, that constitutes the entire input to a single execution of the language translator.

Source Statement: Statements written by a programmer in symbolic terms related to a language translator such as Assembler or COBOL.

Source Statement Library. A collection of books (such as macro definitions) cataloged onto the system tape by the Librarian.

Stacked Job Processing: A technique that permits multiple job definitions to be grouped (stacked) for presentation to the system. This allows the system to automatically process each job in sequence.

Supervisor: A component of the control program. It consists of routines to control the functions of machine interruptions, external interruptions, operator communications, and physical IOCS requests and interruptions.

Symbolic I/O Assignment: A means by which problem programs can refer to an I/O device by a symbolic name. Before a program is executed, Job Control can be used to assign a specific I/O device to that symbolic name.

System Loader: One of the Supervisor routines. It is used to retrieve program phases from the core image library and load them into main storage. System Pack: The pack in which the Basic Operating System resides.

System Residence: The external storage space allocated for storing the basic operating system. It refers to an on-line tape reel that contains the necessary programs required for executing a job on the data processing system.

System Service Programs: Programs that perform portions of the functions of generating the initial basic operating system, generating specialized systems, creating and maintaining the library sections, and loading and editing programs onto the resident tape. These programs are: Linkage Editor and Librarian.

Tape-Resident System: An operating system that uses magnetic tape for on-line storage of system routines.

Throughput: A measure of system efficiency; the rate at which work can be handled by a data processing system.

Transient Area: This is a main storage area (within the Supervisor area) used for temporary storage of transient routines.

Transient Routines: These routines are permanently stored on the system-residence tape and loaded (by the Supervisor) into the transient area when needed for execution.

Undefined Record: A record having an unspecified or unknown length.

Variable-Length Record: A record having a length independent of the length of other records with which it is logically or physically associated. (Contrasted with Fixed-Length Record.)

Volume: That portion of a single unit of storage media that is accessible to a single read/write mechanism. For example, a reel of magnetic tape on a 2400-series magnetic tape drive.

INDEX

READER'S COMMENT FORM

IBM Basic Operating System/360 System Control and System Service Programs (16K Tape) C24-3431-0

- Your comments, accompanied by answers to the following questions, help us produce better publications for your use. If your answer to a question is "No" or requires qualification, please explain in the space provided below. All comments will be handled on a non-confidential basis. Copies of this and other IBM publications can be obtained through IBM Branch Offices.

	Yes	No
• Does this publication meet your needs?	☐	☐
• Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

- What is your occupation? _____
- How do you use this publication?

As an introduction to the subject?	☐	As an instructor in a class?	☐
For advanced knowledge of the subject?	☐	As a student in a class?	☐
For information about operating procedures?	☐	As a reference manual?	☐

Other _____

- Please give specific page and line references with your comments when appropriate. If you wish a reply, be sure to include your name and address.

COMMENTS:

- Thank you for your cooperation. No postage necessary if mailed in the U. S. A.

Staple

Fold

Fold

FIRST CLASS
PERMIT NO. 170
ENDICOTT, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM Corporation
P. O. Box 6
Endicott, N. Y. 13764

Attention: Programming Publications, Dept. 157

Fold

Fold

Cut Along Line

IBM S360 Printed in U. S. A. C24-3431-0

Additional Comments:

IBM

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