

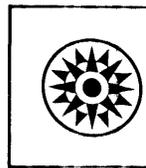
## Systems Reference Library

# IBM System/360 Operating System Operator's Guide

This publication tells how to run the IBM System/360 Operating System. After summarizing how the system works, it describes the three major system types: systems with the primary control program (PCP); systems that provide multiprogramming with a fixed number of tasks (MFT or Option 2); and systems that provide multiprogramming with a variable number of tasks (MVT or Option 4). Instructions are given telling:

- How to start, stop, and restart the system.
- How to control input and output.
- How to control jobs through commands and statements.
- How to understand messages.

General operating techniques are discussed, as well as the meanings of many technical terms.



PREFACE

To fully understand this guide, you should be familiar with the required publication listed below. In addition, you are encouraged to read the recommended publications.

Required Publication

IBM System/360 Operating System: Messages and Codes, Form C28-6631

Recommended Publications

IBM System/360 Operating System: Job Control Language, Form C28-6539

IBM System/360 Operating System: Job Control Language Charts, Form C28-6632

IBM System/360 Operating System: Utilities, Form C28-6586

HOW TO USE THIS GUIDE

- If this is your first reading, read the Introduction, the chapter (1, 2, or 3) describing the type of system you have at your installation, and then Chapters 4, 5, and 6.
- If you have read the book and only want to refresh your memory about a certain procedure, look in the table of contents for the appropriate "how-to-do-it" summary.
- If you want to find the meaning of a technical term, look it up in Chapter 6. If you need more information about the term, look it up in the index and in the table of contents.
- If you have any comments or corrections, fill out and mail the form at the back of the book. Many of the features of this guide are the result of readers' comments on earlier versions.

Sixth Edition (September 1967)

This edition, Form C28-6540-5, makes obsolete all earlier editions and Technical Newsletters. This edition supports release 13 programming by including information on the 2321 data cell drive. It also gives helpful information on operating systems with MVT, and clarifies how to print a SYSOUT tape using the IEFPR1 program and how to issue the SET command. The edition adds information on altering the current PSW. It also corrects errata.

Specifications contained herein are subject to change from time to time. Any such change will be reported in subsequent revisions or Technical Newsletters.

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

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form is provided at the back of this publication for reader's comments. If the form has been removed, comments may be addressed to IBM Corporation, Programming Systems Publications, Department D58, PO Box 390, Poughkeepsie, N. Y. 12602

INTRODUCTION: HOW THE OPERATING SYSTEM WORKS . . . . .	7	Assignment by the Scheduler . . . . .	26
What the Control Program Is . . . . .	7	Assignment by the Operator . . . . .	26
How the Control Program Works . . . . .	7	Allocation Guidelines . . . . .	27
CHAPTER 1: SYSTEMS WITH THE PRIMARY CONTROL PROGRAM . . . . .	9	Operator Commands . . . . .	27
Starting, Stopping, and Restarting the System . . . . .	9	CANCEL -- Terminate Job Immediately . . . . .	28
Starting the System . . . . .	9	DISPLAY -- Cause Console Display . . . . .	28
Initial Program Loading . . . . .	9	MOUNT -- Allocate Device . . . . .	29
Readying the Nucleus . . . . .	11	REPLY -- Reply to Information Request . . . . .	29
Readying the Scheduler . . . . .	11	REQ -- Request Commands . . . . .	29
Stopping the System . . . . .	12	SET -- Set Date, Time, and Location . . . . .	30
Restarting the System . . . . .	12	SHIFT -- Reassign Job Scheduler to Higher Priority Partition . . . . .	30
Controlling Input and Output . . . . .	12	START -- Start System Process . . . . .	31
Input . . . . .	12	STOP -- Stop System Process . . . . .	32
Input Reader . . . . .	12	UNLOAD -- Prepare Volume for Dismounting . . . . .	32
Output . . . . .	12	VARY -- Vary Status of Device . . . . .	32
Output Writer . . . . .	12	Summary of Special MFT Operating Techniques . . . . .	33
Allocating Devices . . . . .	12	How to Start a System With MFT . . . . .	33
Assignment by the Scheduler . . . . .	13	How to Shift the Scheduler . . . . .	34
Assignment by the Operator . . . . .	13	How to Run a Job That Updates System Data Sets . . . . .	34
Allocation Guidelines . . . . .	14	How to Print a SYSOUT Tape . . . . .	34
Operator Commands . . . . .	14	CHAPTER 3: SYSTEMS WITH MVT . . . . .	35
CANCEL -- Terminate Job Immediately . . . . .	14	Starting, Stopping, and Restarting the System . . . . .	35
DISPLAY -- Cause Console Display . . . . .	15	Starting the System . . . . .	37
MOUNT -- Allocate Device . . . . .	15	Initial Program Loading . . . . .	37
REPLY -- Reply to Information Request . . . . .	15	Readying the Nucleus . . . . .	37
REQ -- Request Commands . . . . .	16	Readying the Scheduler . . . . .	37
SET -- Set Date, Time, and Location . . . . .	16	Stopping the System . . . . .	38
START -- Start System Process . . . . .	16	Restarting the System . . . . .	38
STOP -- Stop System Process . . . . .	17	Controlling Input and Output . . . . .	39
UNLOAD -- Prepare Volume for Dismounting . . . . .	17	Input . . . . .	39
VARY -- Vary Status of Device . . . . .	17	Input Readers . . . . .	39
Summary of Special PCP Operating Techniques . . . . .	18	Output . . . . .	39
How to Start a System With the Primary Control Program . . . . .	18	Output Classes . . . . .	39
How to Print a SYSOUT Tape . . . . .	18	Output Writers . . . . .	40
CHAPTER 2: SYSTEMS WITH MFT . . . . .	21	System Log . . . . .	40
Starting, Stopping, and Restarting the System . . . . .	21	Allocating Devices . . . . .	40
Starting the System . . . . .	21	Assignment by the Scheduler . . . . .	42
Initial Program Loading . . . . .	21	Assignment by the Operator . . . . .	42
Readying the Nucleus . . . . .	23	Allocation Guidelines . . . . .	42
Redefining Partitions . . . . .	23	Operator Commands . . . . .	43
Readying the Scheduler . . . . .	24	CANCEL -- Terminate Job Immediately . . . . .	44
Shifting the Scheduler . . . . .	24	DISPLAY -- Cause Console Display . . . . .	45
Planning the Work for each Partition . . . . .	25	HALT -- Prepare for Power-off . . . . .	46
Stopping the System . . . . .	25	HOLD -- Temporarily Suspend Job Selection . . . . .	46
Restarting the System . . . . .	25	LOG -- Store Information in Log . . . . .	46
Controlling Input and Output . . . . .	25	MODIFY -- Alter Output Writer . . . . .	46
Input . . . . .	25	MOUNT -- Allocate Device . . . . .	47
Input Reader . . . . .	25	RELEASE -- Make Job Available for Selection . . . . .	47
Output . . . . .	26	REPLY -- Reply to Information Request . . . . .	48
Output Writer . . . . .	26	RESET -- Change Priority of Job . . . . .	48
Allocating Devices . . . . .	26	SET -- Set Date, Time, and Location . . . . .	48
		START -- Start System Process . . . . .	49

STOP -- Stop System Process . . . . .	50	On the Model 30 . . . . .	63
UNLOAD -- Prepare Volume for Dismounting . . . . .	51	On the Model 40 . . . . .	63
VARY -- Vary Status of Device . . . . .	51	On the Model 50 . . . . .	63
WRITELOG -- Write Out System Log . . . . .	51	On the Model 65 . . . . .	63
Summary of Special MVT Operating Techniques . . . . .	51	On the Model 75 . . . . .	63
How to Start a System With MVT . . . . .	51	Permanent Main Storage Assignments . . . . .	63
How to Specify Job Queue Parameters . . . . .	52	Program Status Word . . . . .	65
How to Determine System Status . . . . .	52	Channel Address Word . . . . .	65
How to Control Jobs Through HOLD and RELEASE . . . . .	53	Channel Command Word . . . . .	65
How to Extract a Job From Tape Input Stream . . . . .	53	Channel Status Word . . . . .	67
How to Run Jobs That Update System Data Sets . . . . .	53	How to Analyze Input/Output Commands . . . . .	67
How to Print a SYSOUT Tape . . . . .	54	How to Use Independent Utility Programs . . . . .	69
Standard IBM Mountable-device Names . . . . .	54	Hardware Debugging Aids . . . . .	69
Start RDR and WTR Device Names . . . . .	54	How to use SEREP . . . . .	71
Universal Character Set (UCS) Procedures . . . . .	54	Creating Volume Labels . . . . .	71
CHAPTER 4: GENERAL OPERATING TECHNIQUES . . . . .	55	Universal Character Set (UCS) Utility Program . . . . .	71
How to Enter a Command Through the IBM 1052 Printer-Keyboard . . . . .	57	Control Statements . . . . .	71
How to Enter a Command Through a Composite Console . . . . .	57	Null Statement . . . . .	73
How to Shift to an Alternate console Device . . . . .	57	Command Statement . . . . .	73
How to Enter a Command into the Input Stream . . . . .	57	Job Statement . . . . .	73
How to Specify System Parameters (all systems) . . . . .	57	Execute Statement . . . . .	73
How to Load a Secondary Nucleus . . . . .	59	Data Definition Statement . . . . .	73
How to Display the Current PSW . . . . .	59	Delimiter Statement . . . . .	73
On the Model 30 . . . . .	61	Additional Operating Techniques . . . . .	75
On the Model 40 . . . . .	61	Model 30 . . . . .	75
On the Model 50 . . . . .	61	Model 40 . . . . .	76
On the Model 65 . . . . .	61	Model 50 . . . . .	77
On the Model 75 . . . . .	61	Model 65 . . . . .	78
How to Bypass a Console Malfunction . . . . .	61	Model 75 . . . . .	79
How to Display Main Storage . . . . .	63	CHAPTER 5: UNDERSTANDING MESSAGES . . . . .	81
		Message Format . . . . .	81
		Message Identification (ID) . . . . .	81
		IBM Standard Message Code . . . . .	81
		Action Indicator . . . . .	82
		Message Text . . . . .	82
		CHAPTER 6: TECHNICAL TERMS AND THEIR MEANINGS . . . . .	83
		INDEX . . . . .	91

FIGURES

Figure 1. How the System Works . . . . .	6	Figure 8. Model 6 Keyboard and Switch Panel . . . . .	60
Figure 2. How the Scheduler Works in Systems With the Primary Control Program . . . . .	10	Figure 9. Model 7 Keyboard and Switch Panel . . . . .	62
Figure 3. How the Scheduler is Shifted in Systems With MFT . . . . .	22	Figure 10. Model 8 Keyboard and SWITCH Panel . . . . .	64
Figure 4. How the Scheduler Works in Systems With MVT . . . . .	36	Figure 11. Model 30 System Control Panel . . . . .	66
Figure 5. Output Process in Systems With MVT . . . . .	41	Figure 12. Model 40 System Control Panel . . . . .	68
Figure 6. Model 3 Keyboard and Switch Panel . . . . .	56	Figure 13. Model 50 System Control Panel . . . . .	70
Figure 7. Model 5 Keyboard and Switch Panel . . . . .	58	Figure 14. Model 65 System Control Panel . . . . .	72
		Figure 15. Model 75 System Control Panel . . . . .	74

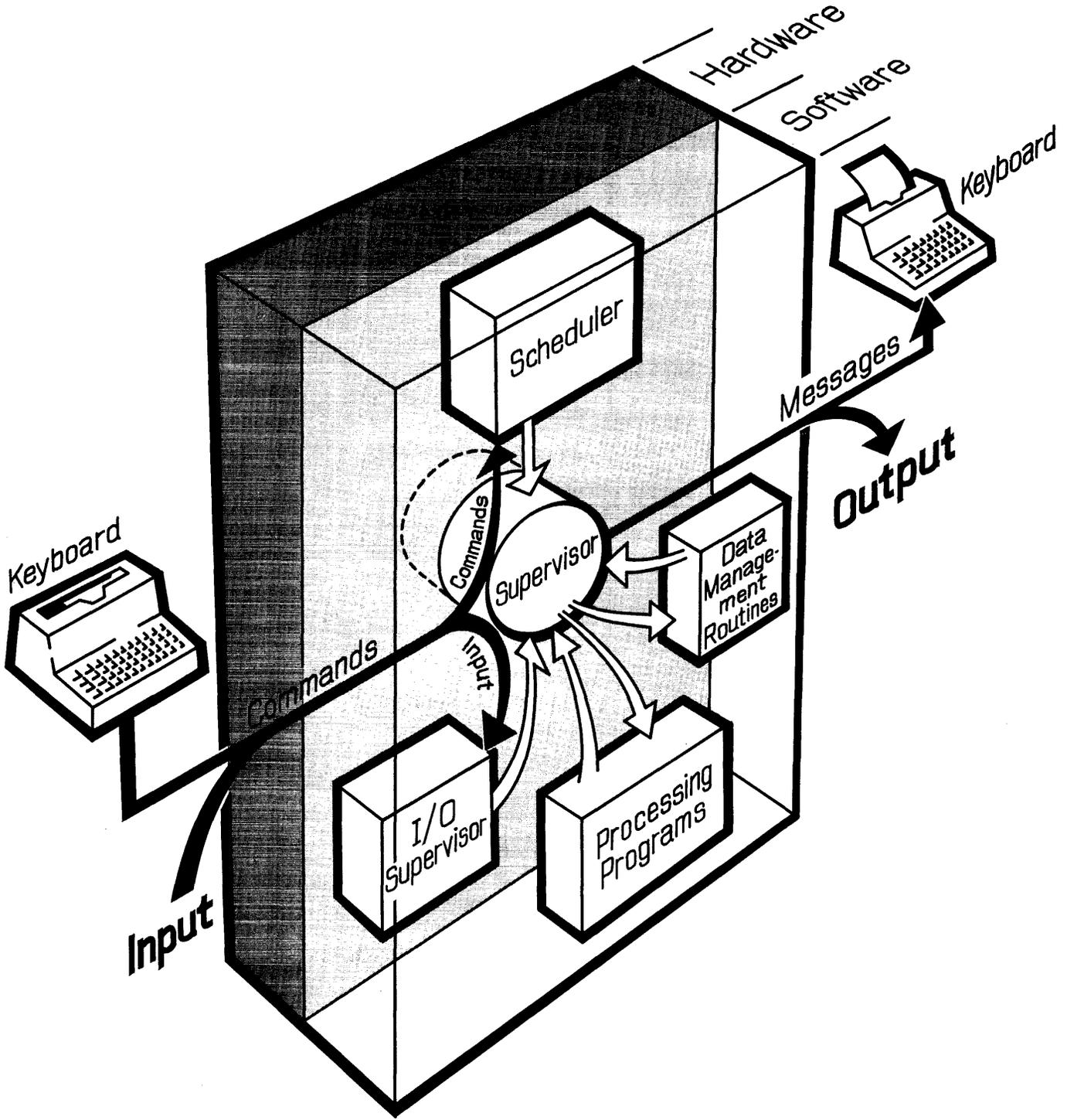


Figure 1. How the System Works

The operating system is an IBM control program and a number of processing programs either written by your programmers or supplied by IBM. As a whole, the operating system helps you handle jobs, by automatically switching from one job to the next.

Figure 1 shows your relationship, through the console device, with the control program parts, the processing programs (which include the rest of the operating system and all the software run with it), and the computer.

As shown in the figure, you communicate with the hardware only through the operating system. Moreover, you communicate with only two parts of the control program - the scheduler and the supervisor.

The supervisor is the key controlling part of the operating system - the only way to use the hardware is through the supervisor. The supervisor itself governs the CPU and main storage. The rest of the hardware - the input/output (I/O) devices - receives instructions from another part of the control program, the I/O supervisor, but only after the I/O supervisor receives control from the supervisor. Similarly, the other major parts of the operating system - the processing programs and the data management routines - are accessible only through the supervisor.

#### WHAT THE CONTROL PROGRAM IS

The control program is that part of the operating system that schedules and organizes the work to be done by the central processing unit, the other hardware, and the software.

The control program supervises the operation of the processing programs, including those written by your programmers. The control program is made up of the scheduler, supervisor, I/O supervisor, and data management routines.

The scheduler - made up of a master scheduler and a job scheduler - analyzes job input and prepares it for execution. The master scheduler analyzes your console commands and directs other control program routines as to what they must do in response. The job scheduler reads the input stream, sets up and starts jobs, stops jobs, and writes job output.

The supervisor coordinates the operation of the whole system. When an interruption occurs - say because you pressed the interrupt button on the console, or because of a machine check - the supervisor determines the cause of the interruption, handles the interruption partially or fully, and determines the next routine to receive control of the CPU when the interruption processing is finished. And the supervisor optionally provides the ability to multiprogram - use main storage for more than one program at a time.

One of the parts of the control program that the supervisor uses frequently is the I/O supervisor - to execute channel programs for example, or to check for tape errors.

Data management controls all I/O operations beyond those covered by the I/O supervisor, and passes data to the I/O supervisor in the form of channel programs.

#### HOW THE CONTROL PROGRAM WORKS

While the system is running, control of the CPU is continually passed between the control program and the processing programs.

After setting up a given job step, the control program passes CPU control to a processing program to perform the work required. A processing program, in turn, gives up CPU control through an interruption to the control program so that the control program can perform some service - handle a Supervisor Call (SVC) instruction, for instance, or an end-of-file condition.

Interruptions, which are an operation of the hardware rather than of a program, ensure that the control program always receives control of the CPU when control program processing is needed.

When you issue a command at the console, the system is interrupted and the control program does what you have commanded. All messages come back to you through the control program.

For the most part, your activities are coordinated with the operations of the two parts of the scheduler - the master scheduler and the job scheduler.

The master scheduler accepts your commands and acts as your agent within the system. It relays system messages to you, starts and stops certain system work at your request, and responds to your inquiries regarding job or system status. Besides handling communication between you and the system, the master scheduler readies the job scheduler at your command.

The job scheduler reads and analyzes job control language (JCL) statements, assigns I/O devices, and starts job steps. Parts of the schedulers, as well as other elements of the control program, differ between the three major types of systems -- systems with the primary control program (PCP), systems that include multiprogramming with a fixed number of tasks (MFT), and systems that include multiprogramming with a variable number of tasks (MVT).

Systems with the primary control program execute jobs as they are read in, one at a time.

Systems with MFT read one to four jobs sequentially into main storage and then

execute them concurrently on a priority basis.

Systems with MVT read jobs in any order, handle them according to priority, and execute them concurrently, letting them share CPU time, main storage, and other system resources.

In the descriptions and illustrations in the next three chapters, the stress is on the main features of each type of system, from the point of view of running the system.

In systems with the primary control program, for example, the most significant aspect is the way the scheduler works. But in systems with MFT, the significance lies in the way main storage is handled by the supervisor. And in systems with MVT, although both the supervisor and the scheduler are more powerful than in other systems, the major operational difference is in the scheduler.

The important point to remember is that each system as a whole works as shown in Figure 1.

In systems with the primary control program, the main parts of the job scheduler are the reader and the initiator (Figure 2). Control is passed from the reader to the initiator, and then to the job step. At any given time, only one of these parts can be in control.

At first, control is given to the reader, which reads in and arranges job control records till data from the current job, or control records for the next job, are found in the input stream.

Control is then passed to the initiator, which assigns devices and starts the job. When the job or job step ends, control is returned to the reader by the initiator, and the process is repeated.

One job at a time is brought from the input stream into main storage. That job can use all of main storage beyond an area set aside for the control program routines that make up the nucleus of the system. No other job can be brought into main storage until the first job is terminated.

#### STARTING, STOPPING, AND RESTARTING THE SYSTEM

System operations are controlled mainly through a console I/O device - the IBM 1052 Printer Keyboard. You give commands to the system, and receive messages from it, through the console.

There may be one console or two, depending on your installation. If you have two, one of them is called the primary console and the other is called the alternate console. If your installation uses a composite console, made up of a card reader and a printer, it will be your only console -- there will be no alternate.

The primary console is active until you shift to the alternate by pressing the INTERRUPT key on the system control panel. Before shifting, give the VARY command to switch the alternate console offline to avoid conflicts with executing programs. For procedures to follow when a console malfunction, see "How to Bypass a Console Malfunction" in Chapter 4.

To give the system a command through an IBM 1052 printer-keyboard, first see if the PROCEED light is on. If it is on, type in your command without further preparation.

If it is not on, press the REQUEST key, wait for the PROCEED light to turn on, and then type in your command. The total number of typed characters per command cannot exceed 126. To mark the end of a command, signal end of block (EOB) by pressing the alternate coding key and the numeric 5 key.

You can cancel a typed line by pressing the alternate coding key and the numeric 0 key.

To give the system a command through a composite console (made up of a card reader and a printer, for example) press the card reader's STOP key, place the command in the reader, and press the START and EOF keys.

You can also enter commands through a SYSIN (system input) stream by using command statements (see the topic "Controlling Jobs"). Each command statement, or group of command statements, must precede a JOB, an EXEC, or a null statement. Commands are accepted as soon as they are read - they are not synchronized with the job in the SYSIN stream.

#### STARTING THE SYSTEM

Starting the system includes initial program loading (IPL), readying the nucleus, and readying the scheduler.

#### Initial Program Loading

Initial program loading is a procedure carried out at the beginning of a shift, after a power-on following an electrical shutoff, after malfunctions that require reloading the control program into main storage, after scheduled maintenance, and as part of switching from one system to another.

Begin initial program loading by selecting the direct-access storage device on which the operating system resides: set the three LOAD UNIT switches on the control panel to the proper channel, control unit and device numbers; then press the LOAD key on the panel.

Pressing the LOAD key turns off the MANUAL light, turns on the LOAD light, and starts reading the IPL program from the input device. After the IPL program is read into lower main storage, control is passed to it, and the LOAD light turns off.

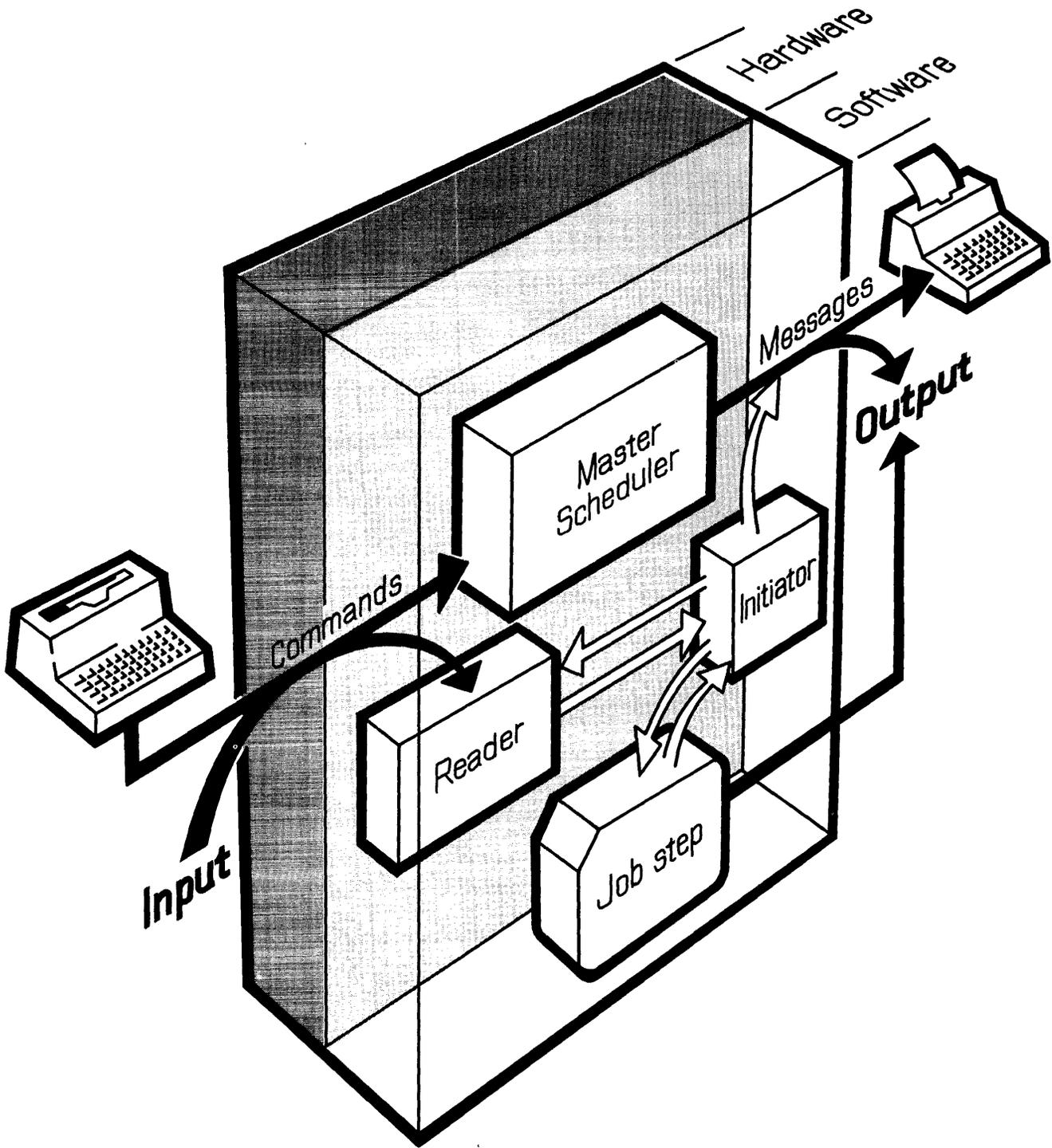


Figure 2. How the Scheduler Works in Systems With the Primary Control Program

If either the reading operation or the passing of control is unsuccessful, the CPU pauses and the LOAD light stays on.

When the IPL program gets control, it loads the nucleus of the control program into main storage. The IPL program loads a standard, or primary, nucleus unless you cause it to load a secondary nucleus. For a description of this procedure, see "How to Load a Secondary Nucleus" in Chapter 4. After the nucleus is loaded, control is given to a nucleus initialization program (NIP).

If the IPL program does not finish successfully, or if I/O errors occur while NIP is running, the WAIT light turns on and an error code is placed in the low-order byte of the program status word (PSW). Whenever the WAIT light turns on without a message, display the PSW, note the error code, and follow the instructions for that code given in the publication IBM System/360 Operating System: Messages, Completion Codes, and Storage Dumps.

#### Readying the Nucleus

The nucleus initialization program (NIP) does general preparatory work for the system. If the communication option was specified at the time the system was generated, you will receive a message, SPECIFY SYSTEM PARAMETERS, requesting any changes.

If you receive this message, your system programmer may ask you to alter one or more options, such as the BLDL option, the RAM option, or the RSVC option. Explanations for the various options, and instructions on how to alter the options, are given in Chapter 4, under the heading "How to Specify System Parameters."

After NIP completes its preparation of the system, it passes control to the master scheduler. You will receive a READY message from the system and the WAIT light will turn on.

If an error other than an I/O error occurs during the running of NIP, the WAIT light turns on, and you will receive a message identifying the error. (No message is sent if the system console is not ready, but a code can be found in the right half of the current PSW as the system waits.)

#### Readying the Scheduler

After initial program loading, issue commands to start the job scheduler, which in turn begins the flow of work through the system.

When the system is ready to run, you will receive a READY message, and the WAIT light will go on. You may then enter commands.

Your first command must be a SET command specifying the date. When the system includes the timer option, the SET command should also give the time of day. Optionally, SET can specify the names of the devices for the input queue and a procedure library (SYS1.PROCLIB), and can also specify input queue formatting. Normally, the formatting parameter (Q=(unitname,F)) can be left out after the first IPL, causing the scheduler to use the input queue as it was formatted earlier.

After issuing SET, you can enter any commands in any order. START RDR and START WTR must be issued if they have not been specified by your installation at system generation time. If your installation has already specified START RDR and START WTR, I/O devices are automatically allocated to an input reader and an output writer, and the commands are written out on the console as if you had keyed them in yourself.

If you want to override an automatic START RDR or START WTR command, wait until after the command appears on the console and then enter your own command manually.

You must enter a START command with no parameters as the last command.

#### Examples:

1. To start a system with automatic START RDR and START WTR:

```
SET DATE=yy.ddd,Q=(unitname,F)
START
```

2. To start a system and to remove an I/O device from the system before processing:

```
SET DATE=yy.ddd,Q=(unitname,F)
START RDR,unitname
START WTR,unitname
VARY unitname,OFFLINE
START
```

3. To start a system with a timer, and to remove two I/O devices before processing:

```
SET DATE=yy.ddd,CLOCK=hh.mm.ss,
Q=(unitname,F)
START RDR,unitname
START WTR,unitname
VARY unitname,OFFLINE
VARY unitname,OFFLINE
START
```

#### STOPPING THE SYSTEM

Stopping the operating system -- to prepare for turning power off, for example, or for loading another system or an independent utility program -- means taking all jobs out of the input stream, and canceling the job in progress or waiting for it to end.

#### RESTARTING THE SYSTEM

Follow the same steps you took in starting the system, but leave out the formatting parameter, Q=(unitname,F), when you issue the SET command.

#### CONTROLLING INPUT AND OUTPUT

This section describes how to work with the operating system in controlling input and output, particularly in the area of allocating or assigning devices to jobs.

#### INPUT

A system input (SYSIN) stream is made up of job control language (JCL) statements and problem program data routed by the control program to their destinations within the system. Sets of input data entering the system through the SYSIN stream are called SYSIN data sets. Input data can also be read directly by the problem program and not be a part of a SYSIN stream.

#### Input Reader

An input reader is that part of the scheduler that reads a system input stream from a single device. You assign an input device to an input reader by issuing a START RDR command.

Input is in the form of cards or 80-character unblocked records (card images) on tape.

There is only one input reader in the system, but you can change the source of the SYSIN stream from one unit to another by another START RDR command naming the new unit. At the end of the current job, the input reader will start reading from the new unit.

Systems with the primary control program handle only one job at a time. The processing of JCL statements halts temporarily when a SYSIN data set is met in the input stream. The unit handling the SYSIN stream is then assigned to the job step. The job step reads the data set directly from the

unit until a JCL delimiter statement (/\*) is met. The delimiter statement marks the end of data for the problem program, separating it from the JCL statements for the next job step.

#### OUTPUT

A system output (SYSOUT) stream consists of system messages and problem program output data sets routed by the control program to a common output device. Problem program data sets that leave the system through the output stream are called SYSOUT data sets. Output data can also be written directly by a problem program and not be a part of a SYSOUT stream.

#### Output Writers

An output writer is that part of the scheduler that writes a single class of output to a single device. A maximum of eight output writers can be active at any one time, each writing one class to one device. Each output class is designated by a one-character class name.

Valid class names include the letters A through Z, and the numbers 0 through 9. Class A is the system output class. An output device must be active for class A at all times.

An output device is assigned to an output class by issuing a START WTR command with a classname. The device is unassigned from the output class by issuing either a STOP WTR command or a START WTR command for that class. The starting of a new writer for an output class automatically stops the writer assigned earlier to that device and class.

System messages are written either before the job step is started or after it ends. When a SYSOUT data set is to be written, the job scheduler allocates the unit handling the SYSOUT stream to the job step, which then writes directly to the unit.

#### ALLOCATING DEVICES

Device allocation is the assignment of I/O devices for use by a job step or the system.

Device allocations are made in response to requests from three sources:

- Data definition statements. These may specify the I/O requirements of a job step in the system.

- System generation statements. These may request fixed assignments for system processes, such as the automatic starting of an input reader.
- Operator commands. These may request assignments for system processes, or may modify assignments made when the system was generated.

### Assignment by the Scheduler

The job scheduler assigns I/O resources needed for the data sets of a job step. A programmer specifies these input/output requirements in his data definition statements. Using this information, the job scheduler attempts to provide overlapped operation, conserve input/output resources, and recognize items that increase input/output efficiency. The job scheduler then allocates I/O devices directly to the job step.

### Assignment by the Operator

You are responsible for device assignments made for the starting of input readers and output writers. If assignments were specified at system generation time, you do not have to respecify them, unless you wish to make a modification. All of your assignment requests are made through operator commands.

Your ability to switch input/output devices to online or offline status lets you modify their allocation. Online devices can be assigned to problem programs, while offline devices cannot be. Your main reason for placing a device offline is to reserve it for preventive maintenance. To switch a device's status, enter a VARY command indicating the device and the desired status.

By issuing the MOUNT command, you can cause an input/output device to be assigned to those job steps that require the particular volume mounted on it. For example, you might give a MOUNT command when you know that a volume will soon be used by many independent jobs.

In systems with the automatic volume recognition option, you can mount volumes on online devices that are not ready, thus anticipating the later needs of jobs you are scheduling.

Volume Mounting: In most installations, your role with respect to I/O devices is to mount and dismount volumes.

The job scheduler, using information from data definition statements, determines the input/output resources to be assigned to a job and the volumes that are required. If these volumes are not mounted, the job scheduler writes you a mounting message. Each message states that either a specific volume or a scratch volume is to be mounted. Mount the requested volume and press the START key on the device to continue processing.

You should never mount a blank tape volume. The system checks for a tape label and the absence of data causes the whole volume to be scanned for a data record. If an unlabeled tape is required, a tape mark should be written to avoid unnecessary scanning. For a description of a program you can use to create labels on standard label tapes, see the section on the program IEHINITT in the publication IBM System/360 Operating System: Utilities.

After you mount the volume and ready the drive, the system reads the volume label. If an incorrect volume is mounted, the system repeats the mounting message and unloads the incorrect volume, if possible (some devices, such as the 2311, can only be unloaded manually). If a request was made for a tape volume without a standard label and if the volume mounted does not have a standard label, that volume will be accepted. The volume is treated as unlabeled, or as a volume labeled with non-standard labels, according to the DD statement.

The following volume mounting options can be selected at system generation time:

- Imperative mount. Mounting messages are written when the job step requiring the volumes is started.
- Automatic volume recognition. You take the initiative and mount labeled volumes on any unused drives. The system recognizes and remembers these volumes and assigns the drives to later job steps.

If your system has the automatic volume recognition feature, mount volumes you want the system to find for the first job at IPL before issuing the START command. Also before issuing START, be sure that all offline devices are known to the system by using the VARY OFFLINE command. After the first job, you can mount ahead for several jobs at a time. In addition, the system may ask you to mount other volumes, and you can mount these on any appropriate online devices that are not ready. (Do not unload any units - you can only mount on units unloaded by the system.)

Automatic volume recognition handles nine-track tape, seven-track tape, and 2311 and 2314 direct-access devices. The density for seven-track tape is set at 200, 556, or 800 bytes per inch at the time the system is generated.

When volumes are to be dismounted, the system unloads the devices, if possible, and writes you messages identifying the volumes being unloaded.

2321 Note: When you receive a mounting message for a 2321 data cell, press the RESET button on the device if the requested cell is already positioned properly. If you have to open the door on the unit to position the cell, then you do not have to press RESET -- closing the door performs the same function in this case.

### Allocation Guidelines

When the scheduler cannot satisfy requests for allocation from available (online) devices, it sends you a message and a list of offline devices. You then either cancel the job or make an offline device available by replying to the message with a three-character device name.

Use the MOUNT command to reserve a volume on a device, when you know that several jobs are going to need that volume.

Volumes reserved through a MOUNT command are not dismounted by the system until an UNLOAD command is given, causing the system to unload the volume.

Device Names: When referring to I/O devices in the unitname parameters of operator commands, you must use the unique unit names assigned to each device. Symbolic device names of one to eight alphanumeric characters may be defined by your installation, but these are for use by your programmers in their data definition statements. Do not use symbolic names in operator commands.

### OPERATOR COMMANDS

This section contains a description of the commands you use to give control information to the operating system. The formats, functions, parameters, and options of the commands are included. For convenience, the commands are presented in alphabetical order.

The commands usable in systems with the primary control program are:

CANCEL	SET
DISPLAY	START
MOUNT	STOP
REPLY	UNLOAD
REQ	VARY

Console commands, other than SET and START, are accepted whenever you issue them. To issue a SET or START command, first issue a REQ command. A REQ command makes the system pause and issue a message requesting further commands at the end of a job step. Any commands issued in response to such a message must be followed by a

START command with no parameters. This command gives control to the reader to start the next job step.

The following conventions are used in illustrating the format of commands:

- Required letters (those shown in upper case) must be entered, but can be entered in either upper or lower case.
- Lower-case letters indicate that a parameter must be substituted.
- Dotted lines ... (indicating a series of terms), brackets [ ], and braces { } are not entered.
- Entries within brackets [ ] are optional.
- Entries within braces { } are required - you must select one.
- Numbers and punctuation marks (other than dotted lines, brackets, and braces) must be entered as shown.

Command formats are essentially free form, but one or more blanks must follow the operation field. Commands cannot occupy more than one line. For example, if a command is entered through a card reader, it may not be more than 80 characters in length.

If comments on commands are necessary, they should appear to the right of the operand field and be separated from it by at least one blank. If the operand field is null, a comma followed by at least one blank indicates that comments will follow.

### CANCEL -- Terminate Job Immediately

The CANCEL command is used to immediately terminate the scheduling or execution of a job. Optionally, you may request that an abnormal-end-of-task storage dump be taken if the command is received while the job is running. This command is always executed as soon as it is received.

If you enter a CANCEL command for a job that is neither running nor in the process of being scheduled, you will be informed that the command cannot be executed.

Operation	Operand
CANCEL	jobname[,DUMP]

jobname

specifies the name of the job to be terminated. The maximum length of a job name is eight characters.

DUMP

specifies that an abnormal-end-of-task storage dump is to be taken if a step of the job is being executed when the command is received. If the programmer has put in the SYSABEND data definition statement, a full dump is taken. If he has not included this card, an indicative (partial) dump is taken.

Note: You may be asked by your programmers not to use the CANCEL command on certain jobs. These jobs alter data sets containing information vital to the system -- canceling the jobs might make the data unusable.

DISPLAY -- Cause Console Display

The DISPLAY command is used primarily to cause a console display of the name of each job at the time it is initiated, and at the time it is terminated. (If a job is terminated due to unusual circumstances, you will receive a message even if you have not used the DISPLAY command.) The command provides you with job name information needed for effective use of the CANCEL command and, together with system messages, keeps you informed of which job is currently being executed.

Operation	Operand
DISPLAY	{ JOBNAMES } { STATUS }

JOBNAMES

specifies that the name of each job is to be displayed both when the job starts and when it terminates, and that unit record allocation is to be displayed when the step starts.

STATUS

specifies that the data set names and volume serial numbers of data sets with dispositions of KEEP, CATLG, or UNCATLG, are to be displayed on the console at step termination and job termination.

MOUNT -- Allocate Device

The MOUNT command is used to allow allocation of an input/output device to all job steps that require a particular volume, without intervening dismountings and remountings of that volume.

The required volume should be mounted immediately after the MOUNT command is issued. A VOL parameter is not used with labeled volumes. The system automatically uses the value read from the label.

Operation	Operand
MOUNT	unitname [,VOL=(NL,serial)]  { STORAGE } { PUBLIC } { PRIVATE }

unitname

specifies the name of the input/output device to be allocated. Unitname must specify a device that has been unloaded by the system.

VOL=(NL,serial)

specifies that the volume does not have standard labels. The alphanumeric serial number, up to six characters long, is used for allocation references. This parameter must not be used for direct-access volumes.

USE=STORAGE or PUBLIC or PRIVATE

specifies that a direct-access volume will be used as either a storage volume or a public volume or a private volume. If this operand is not used, the system treats the volume as a private volume. A storage volume is the most freely allocated kind of volume, open to use by the largest variety of data sets, temporary or non-temporary. Slightly restricted is a public volume, which can be allocated freely for temporary data sets, but which must be specified by volume serial number to be allocated to non-temporary data sets. A private volume is the least freely allocated kind of volume -- it is allocated only if its volume serial number is specified.

2321 Note: When issuing this command for a 2321 data cell, unitname must contain a bin number in addition to the channel, control unit, and device. For example, 263/8 is the unitname for bin number 8 of a 2321 with a device number of 3, a control unit number of 6, and a channel number of 2.

REPLY -- Reply to Information Request

The REPLY command is used to reply to messages from the operating system and from problem programs that request information.

The message requesting the reply must be directly followed by the REPLY command. No other messages are printed until the reply has been entered.

Operation	Operand
REPLY	id,'text'



Any sequence of commands submitted in response to a READY message must be followed by a START command with no parameters.

If the output writer is associated with a tape unit, the volume mounted must have a standard label.

Operation	Operand
START	{RDR}, unitname, {WTR} [volumeserial], [parmvalue], [DSN=datasetname,] [FILESEQ=file sequence number]

RDR specifies that an input reader is to be started.

WTR specifies that an output writer is to be started.

unitname specifies the name of the unit record or magnetic tape input/output device associated with the input reader or output writer that is to be started.

volumeserial specifies the alphameric serial number, up to six characters long, of a magnetic tape volume. If this parameter is specified, label checking is performed.

parmvalue specifies either an up-to-eight-character name of a job in the input stream or a one-character alphameric output class name. A job name is used only with RDR: when starting an input reader, giving the job name causes forward spacing through the input stream until the named job is found. An output class name is used only with WTR: giving a class name indicates the class of output the writer is to handle; if no class is specified, the writer is assigned to class A.

DSN=datasetname specifies the name of the data set associated with the input reader or output writer. The maximum length of a data set name is 44 characters. If this parameter is not specified, the data set name SYSIN is assumed for the reader.

FILESEQ=file sequence number specifies the file sequence number, up to four digits long, of a data set on a magnetic tape volume. This parameter is optional and is used only with RDR when unitname designates a magnetic tape device.

STOP -- Stop System Process

The STOP command is used to stop a console display of job names or of data set names, or to stop an output writer.

Operation	Operand
STOP	{JOBNAMES {WTR, unitname} {STATUS}

JOBNAMES specifies that a console display of the names of jobs, initiated by the JOBNAMES parameter of the DISPLAY command, is to be terminated. For more information about JOBNAMES, see the discussion of the DISPLAY command.

WTR,unitname specifies that the output writer using the named unit is to be stopped by the system. This operand will not stop a writer assigned to class A -- only a new START WTR to class A will cause the presently active one to stop.

STATUS specifies the discontinuance of a console display at step termination and job termination of the names and volume serial numbers of data sets with dispositions of KEEP, CATLG, or UNCATLG.

UNLOAD -- Prepare Volume for Dismounting

This command is normally used to remove a volume previously mounted in response to a MOUNT command.

The UNLOAD command causes a volume on an input/output device to be prepared for dismounting. When the volume is ready to be dismounted, you will receive a message. (The message may not be received until the current job is completed.)

Operation	Operand
UNLOAD	unitname

**unitname**

specifies the unit address of the input/output device to be prepared for dismounting.

2321 Note: When issuing this command for a 2321 data cell, unitname must contain a bin number in addition to the channel, control unit, and device. For example, 263/8 is the unitname for bin number 8 of a 2321 with a device number of 3, a control unit number of 6, and a channel number of 2.

VARY -- Vary Status of Device

The VARY command is used to place an input/output device (other than a communications line) into an online or offline status.

Operation	Operand
VARY	unitname {, ONLINE } {, OFFLINE}

**unitname**

specifies the unit address name of the input/output device whose status is to be changed.

**ONLINE**

specifies that the device is to be made available for allocation by the job scheduler to problem programs.

**OFFLINE**

specifies that the device is to be removed from the recognition of the job scheduler, and that any further allocation of the device to problem programs is to be prevented. If the device is in use (allocated to a problem program or to an input reader or output writer), the status is not changed until the current user is finished with the device. When the status is changed to offline, you will receive a message. A device can be removed from the offline status only by a subsequent VARY command.

2321 Note: To vary the status of an entire 2321 data cell, use its three-character unit address -- 263, for example. To vary the status of a particular 2321 bin, use its five-character unit address -- 263/8, for example, where the 8 is the number of the particular bin being addressed.

SUMMARY OF SPECIAL PCP OPERATING TECHNIQUES

This section covers special procedures that do not apply to all three versions of the operating system. After reading this section, skip to Chapter 4, which describes general techniques.

HOW TO START A SYSTEM WITH THE PRIMARY CONTROL PROGRAM

- Set the LOAD UNIT switches to the channel, control unit, and device to address the system residence volume.
- Press the LOAD key and wait for the LOAD light to go out. If it does not go out after a short period of time, make sure the unit is ready, and then press the LOAD key again.
- Wait for a READY message and the WAIT light to go on. If the WAIT light goes on without a READY message, display the current PSW, and follow the directions for the error code you will find in the right half of the PSW. If the wait light goes on but there is no error code in the PSW, see if you have mounted and addressed the system residence volume correctly.
- After you receive a READY message, enter a SET command with the parameters your installation requires.
- Enter other commands that may be required by your installation, such as DISPLAY JOBNAMES and START. If your installation specified automatic START commands, the following will appear on the console:  
  
START RDR,unitname  
START WTR,unitname
- Enter a START command with no parameters as your last command.

When the nucleus is initialized, you may be required by your installation to alter some optional features. For descriptions of these features and instructions on how to alter them, see "How to Specify System Parameters" in Chapter 4.

HOW TO PRINT A SYSOUT TAPE

In systems with the primary control program, execute the program called IEFPRF to print out a tape prepared as SYSOUT. Punch the following cards and schedule them as a job:

//xxx	JOB	xxx		The variables are:
//xxx	EXEC	PGM=IEFPRT		
//PRINTAP	DD	DSNAME=SYSOUT,	c	
//		UNIT=tapename,	c	xxx
//		DISP=OLD,	c	which is defined by your installation.
//		VOLUME=(SER=(serial 1,	c	
//		serial 2,...)),	c	
//		DCB=DEN=d		tapename
//PRINTER	DD	UNIT=printname		identifies the tape drive or drives
				you wish to use for input.

serial 1, serial 2, etc.  
are the serial numbers of the tape  
volumes you wish printed out.

d  
is a one-digit number specifying the  
density of the tape(s) to be printed.

printname  
is a unit address. It must specify a  
printer. If only one printer is in  
the system and it is being used for  
system output, printname should not be  
used. Instead, the last card would  
be:

//PRINTER DD SYSOUT=A

IEFPRT prints the entire contents of the  
SYSOUT tape or tapes, stopping when the  
special end record is encountered. This  
record is the only record of its data set  
and contains END OF OUTPUT as its first 13  
characters.

Note: The system may not be able to print  
the contents of the SYSOUT tape or tapes if  
your programmers have specified the density  
for their SYSOUT data sets in DD statements  
or in DCB macro instructions.



In systems with MFT, main storage beyond the nucleus area is separated into from one to four areas called partitions. The number and size of these partitions are set when the system is generated, but you can redefine the sizes and reduce the number of partitions when you start the system. Each partition is treated by the supervisor almost as though it were the main storage of a separate computer -- one job is read and scheduled for execution in each partition, and only that job uses that partition until it is terminated.

The partition at the end of main storage with the highest addresses is called P0 and is the highest-priority partition. Any job read into P0 is given top priority whenever the system must decide which job in main storage is to receive control next.

Similarly, the partitions between P0 and the nucleus area -- P1, P2, and P3, as shown in Figure 3 -- are the second-highest, third-highest, and lowest-priority partitions, and their priorities are given to the jobs that are read into them.

#### STARTING, STOPPING, AND RESTARTING THE SYSTEM

System operations are controlled mainly through a console I/O device - the IBM 1052 Printer-Keyboard. You give commands to the system, and receive messages from it, through the console.

There may be one console or two, depending on your installation. If you have two, one of them is called the primary console and the other is called the alternate console. Either console may be a composite console, made up of a card reader and a printer.

The primary console is active until you shift to the alternate by pressing the INTERRUPT key on the system control panel. Before shifting, give the VARY command to switch the alternate console offline to avoid conflicts with executing programs. For procedures to follow when a console malfunctions, see "How to Bypass a Console Malfunction" in Chapter 4.

To give the system a command through an IBM 1052 Printer-Keyboard, first see if the PROCEED light is on. If it is on, type in your command without further preparation. If it is not on, press the REQUEST key, wait for the PROCEED light to turn on, and

then type in your command. The total number of typed characters per command cannot exceed 126. To mark the end of a command, signal end of block (EOB) by pressing the alternate coding key and the numeric 5 key.

You can cancel a typed line by pressing the alternate coding key and the numeric 0 key.

To give the system a command through a composite console (made up of a card reader and a printer, for example) press the card reader's STOP key, place the command in the reader, and press the START and EOF keys.

You can also enter commands through a SYSIN (system input) stream by using command statements (see the topic "Controlling Jobs"). Each command statement, or group of command statements, must precede a JOB, an EXEC, or a null statement. Commands are accepted as soon as they are read - they are not synchronized with the jobs in the SYSIN stream.

#### STARTING THE SYSTEM

Starting the system includes initial program loading (IPL), readying the nucleus, and readying the scheduler.

#### Initial Program Loading

Initial program loading is a procedure carried out at the beginning of a shift, after a power-on following an electrical shutoff, after malfunctions that require reloading the control program into main storage, after scheduled maintenance, and as part of switching from one system to another.

Begin initial program loading by selecting the direct-access storage device on which the operating system resides: set the three LOAD UNIT switches on the control panel to the proper channel, control unit and device numbers; then press the LOAD key on the panel.

Pressing the LOAD key turns off the MANUAL light, turns on the LOAD light, and starts reading the IPL program from the input device. After the IPL program is read into lower main storage, control is passed to it, and the LOAD light turns off. If either the reading operation or the passing of control is unsuccessful, the CPU pauses and the LOAD light stays on.

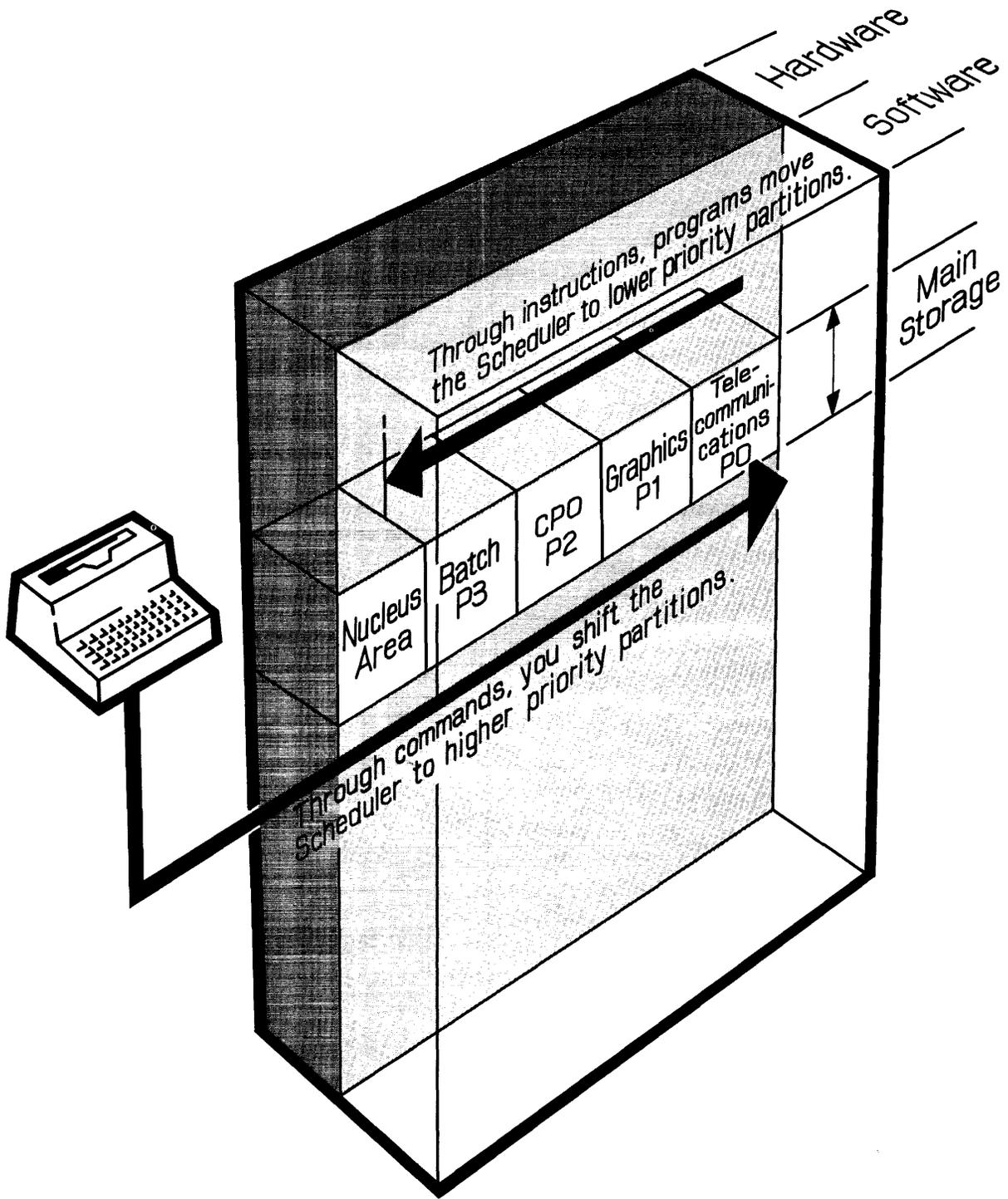


Figure 3. How the Scheduler is Shifted in Systems With MFT

When the IPL program gets control, it loads the nucleus of the control program into main storage. The IPL program loads a standard, or primary, nucleus unless you cause it to load a secondary nucleus. For a description of this procedure, see "How to Load a Secondary Nucleus" in Chapter 4. After the nucleus is loaded, control is given to a nucleus initialization program (NIP).

If the IPL program does not finish successfully, or if I/O errors occur while NIP is running, the WAIT light turns on and an error code is placed in the low-order byte of the program status word (PSW). Whenever the WAIT light turns on without a message, display the PSW, note the error code, and follow the instructions for that code given in the publication IBM System/360 Operating System: Messages, Completion Codes, and Storage Dumps.

### Readying the Nucleus

The nucleus initialization program (NIP) does general preparatory work for the system. If the communication option was specified at the time the system was generated, you will receive a message, SPECIFY SYSTEM PARAMETERS, requesting any changes.

If you receive this message, your system programmer may ask you to alter one or more options, such as the BLDL option, the RAM option, or the RSVC option. Explanations for the various options, and instructions on how to alter the options, are given in Chapter 4, under the heading "How to Specify System Parameters."

If an error other than an I/O error occurs during the running of NIP, the WAIT light turns on, and you will receive a message identifying the error. (No message is sent if the system console is not ready, but a code can be found in the right half of the current PSW as the system waits.)

In systems with MFT, you will receive a LIST PARTITION SIZES? message during the operation of NIP. You can then redefine the size of the partitions.

### Redefining Partitions

At the time the nucleus is initialized, you can redefine the number and size of partitions in a system with MFT. The number of partitions can be reduced from that defined at system generation.

The size you enter for the lowest priority partition -- the one with the largest partition number -- must be at

least that of the scheduler used in the system. The size you enter for each of the other partitions must be at least 6144. The total size of all partitions and the nucleus cannot exceed the amount of main storage available.

The nucleus initialization program issues a series of messages, some of which require your replies. These replies specify whether partitions are to be redefined, the values of the new parameters, and the end of redefinition information.

When initial program loading has been completed, NIP issues the message LIST PARTITION SIZES? Reply to this message with REPLY id, 'YES' if the partition sizes are to be listed, or with REPLY id, 'NO' if they are not.

Generally, use the YES reply even when no redefinition is intended. This reply causes recording on the console device of the number and size of partitions, and the amount of unused storage space given to the lowest priority partition.

If you enter the NO reply, the READY message is issued, and you proceed with system initialization. If you enter the YES reply, the system issues:

P0=number P1=number P2=number P3=number  
EXCESS=number BYTES WILL BE GIVEN TO LOWEST  
PRIORITY PARTITION

This message is followed immediately by the message INDICATE CHANGE, LIST, OR END. At this time you can enter REPLY id, 'END' and signal end-of-block (EOB) if the partitions are not to be redefined. The system responds to END by issuing READY. Alternatively, you can redefine the partitions and, if required, obtain a listing of the newly defined partitions.

Entering the Partition Size: To redefine the size of partitions, respond to the message INDICATE CHANGE, LIST, OR END by entering the new partition sizes in the form REPLY id, 'P<sub>n</sub>=x', up to 80 characters, followed by the signal EOB. The number, n, of a partition is specified in decimal. The number, x, of bytes in a partition is specified in decimal, and should be at least as large as the scheduler used in the system.

In systems with storage protection, the partition size specification is rounded to the next higher multiple of 2048 bytes by the nucleus initialization program. Otherwise, the partition size specification is rounded to the next higher multiple of 8 bytes, if necessary.

To reduce the total number of partitions below that set at the time the system was generated, first redefine the sizes of the partitions you want in the system, and enter REPLY id, 'P<sub>n</sub>=LAST END' and the signal EOB. The system sets the size of P<sub>n</sub> equal to its former size plus the size of the remaining storage space.

Follow the redefinition by the entry REPLY id, 'P<sub>n</sub>=LAST LIST' and signal EOB, if a list of the partitions and sizes is needed. Then enter REPLY id, 'END' and signal EOB to terminate redefinition.

Error Recovery: In systems with MFT, you can cancel errors made before an entry is complete by holding down the alternate coding key and pressing the numeric 0 key. Errors in redefinition recognized before the END entry are canceled simply by entering the correct definition. Errors in entry format cause the system to issue the message INVALID PARAMETER/FORMAT--RESPECIFY.

Partition size errors that cause the total amount of storage space requested to exceed the amount of available space cause the system to issue the message TOTAL SIZE OF PARTITIONS IS x BYTES TOO LARGE FOR STORAGE. Redefine the partition to reduce the total storage request. After redefinition (and, optionally, a request for a list), enter END and signal EOB; the system responds by issuing the READY message.

### Readying the Scheduler

After the nucleus is initialized, issue commands to start the job scheduler, which in turn begins the flow of work through the system.

When the system is ready to run, you will receive a READY message, and the WAIT light will go on. You may then enter commands.

Your first command must be a SET command specifying the date. When the system includes the timer option, the SET command should also give the time of day. Optionally, SET can specify the names of the devices for the input queue and a procedure library (SYS1.PROCLIB), and can also specify input queue formatting. Normally, the formatting parameter (Q=(unitname,F)) can be left out after the first IPL, causing the scheduler to use the input queue as it was formatted earlier.

After issuing SET, you can enter any commands in any order. START RDR and START WTR must be issued if they have not been given by your installation at system generation time. If your installation has already specified START RDR and START WTR,

I/O devices are automatically allocated to an input reader and an output writer, and the commands are written out on the console as if you had keyed them in yourself.

If you want to override an automatic START RDR or START WTR command, wait until after the command appears on the console and then enter your own command manually.

You must enter a START command with no parameters as the last command.

The DISPLAY command with the JOB NAMES parameter should be used as one of the initialization commands. If it is not used, you will have great difficulty in reconstructing the day's work from console messages to find out which jobs are active at any given time. In addition, if you use DISPLAY JOB NAMES, you will have the job names available if you have to cancel any jobs.

### Examples:

1. To start a system with automatic START RDR and START WTR:

```
SET DATE=yy.ddd,Q=(unitname,F)
DISPLAY JOB NAMES
START
```

2. To start a system and to remove an I/O device from the system before processing:

```
SET DATE=yy.ddd,Q=(unitname,F)
START RDR,unitname
START WTR,unitname
VARY unitname,OFFLINE
DISPLAY JOB NAMES
START
```

3. To start a system with a timer, and to remove two I/O devices before processing:

```
SET DATE=yy.ddd,CLOCK=hh.mm.ss,
    Q=(unitname,F)
START RDR,unitname
START WTR,unitname
VARY unitname,OFFLINE
VARY unitname,OFFLINE
DISPLAY JOB NAMES
START
```

### Shifting the Scheduler

When the system is started, the scheduler is assigned to P0. The scheduler reads the first job in the input stream into P0. The first job is executed in that partition, as well as all other jobs, if none of their programs issues instructions to the system to move the scheduler to the next lower partition.

On receiving the proper instructions from a program in the highest-priority partition, the system moves the scheduler to the next partition. The scheduler reads the next job in the input stream into that partition, and starts the job.

As the system receives instructions from programs within jobs, it moves the scheduler to the third and fourth partitions, thus filling main storage with jobs.

Only one job can be executed at any one instant; assume, for the moment, that it is the one in the lowest-priority partition. When any of the higher-priority jobs are ready to run again, the system passes control to them in order of their priority, leaving the scheduler in the lowest-priority partition.

The system does not shift the scheduler to higher-priority partitions unless you tell it to do so by issuing the SHIFT command.

A delay between the time when a job completes its work and the time when the system removes it from main storage is possible in these systems. This can happen if the job that has reached completion is in a partition other than the one the scheduler is in. If this happens, the system sends you a message that the completed job is waiting to be terminated.

### Planning the Work for each Partition

Try to plan as much as possible of the day's work in relation to the best partition assignment for each job.

In general, run those jobs known to rely heavily on I/O operations in one of the higher-priority partitions; run jobs known to rely heavily on central processing unit operations in one of the lower-priority partitions.

If a telecommunications job is to be run, you can make it the first job in the input stream, so that it will be scheduled for execution in the highest-priority partition. Graphics jobs and CPO (concurrent peripheral operations) can be run in the next-highest-priority partitions, if they are to be run in the same system as a telecommunications job.

If no telecommunications job is to be run, graphics and CPO can be scheduled for execution in the highest and second-highest-priority partitions. Put batched jobs into the lowest-priority partition.

### STOPPING THE SYSTEM

Stopping the operating system -- to prepare for turning power off, for example, or for loading another system or an independent utility program -- means taking all jobs out of the input stream, shifting the scheduler to the highest priority partition, canceling each job one at a time or waiting for all the jobs to end normally. If any of the partitions contain teleprocessing jobs that do not normally end by themselves, you will have to cancel them.

### RESTARTING THE SYSTEM

Follow the same steps you took in starting the system, but leave out the formatting parameter, Q=(unitname,F), when you issue the SET command.

### CONTROLLING INPUT AND OUTPUT

This section describes how to work with the operating system in controlling input and output, particularly in the area of allocating or assigning devices to jobs.

#### INPUT

A system input (SYSIN) stream is made up of job control language (JCL) statements and problem program data routed by the control program to their destinations within the system. Sets of input data entering the system through the SYSIN stream are called SYSIN data sets. Input data can also be read directly by the problem program and not be a part of a SYSIN stream.

Problem program input data should be read directly by the problem program rather than have such data a part of the SYSIN stream, unless the problem program is in the lowest-priority partition.

#### Input Reader

An input reader is that part of the scheduler that reads a system input stream from a single device. You assign an input device to an input reader by issuing a START RDR command.

The input stream source can be changed to another unit by issuing another START RDR command specifying the new unit. The processing of job control statements is suspended when a DD\* or DD DATA statement is found in the input stream.

## OUTPUT

A system output (SYSOUT) stream consists of system messages and problem program output data sets routed by the control program to a common output device. Problem program data sets that leave the system through the output stream are called SYSOUT data sets. Output data can also be written directly by a problem program and not be a part of a SYSOUT stream.

### Output Writers

An output writer is that part of the scheduler that writes a single class of output to a single device. A maximum of eight output writers can be active at any one time, each writing one class to one device. Each output class is designated by a one-character class name.

Valid class names include the letters A through Z, and the numbers 0 through 9. Class A is the system output class. An output device must be active for class A at all times. Only problem programs in the lowest priority partition can use class A.

An output device is assigned to an output class by issuing a START WTR command with a classname. The device is unassigned from the output class by issuing either a STOP WTR command or a START WTR command for that class. The starting of a new writer for an output class automatically stops the writer assigned earlier to that device and class.

Although system messages are written either before or after the execution of the job step to which they refer, their writing may take place concurrently with the execution of jobs in other partitions.

When a SYSOUT data set is to be written, the job scheduler allocates the unit handling the SYSOUT stream to the job step, which then writes directly to the unit.

## ALLOCATING DEVICES

Device allocation is the assignment of I/O devices for use by job steps or the system. If a job step is using a tape drive, no other job step may use the tape drive until the first job step is finished - the entire device is assigned. If the

job step is writing on a direct-access volume, however, it may only be necessary to assign a particular set of tracks; data sets for many job steps can be on the same volume.

Device allocations are made in response to requests from three sources:

- Data definition statements. These may specify the I/O requirements of the job steps in the system.
- System generation statements. These may request fixed assignments for system processes, such as the automatic starting of an input reader.
- Operator commands. These may request assignments for system processes, or may modify assignments made when the system was generated.

### Assignment by the Scheduler

The job scheduler assigns I/O resources needed for the data sets of a job step. A programmer specifies these input/output requirements in his data definition statements. Using this information, the job scheduler attempts to provide overlapped operation, conserve input/output resources, and recognize items that increase input/output efficiency. The job scheduler then allocates I/O devices directly to the job step.

Although in systems without the automatic volume recognition option (discussed later) all device allocations to job steps are made by the system, you can indirectly influence such allocations as those described in the next section.

### Assignment by the Operator

You are responsible for device assignments made for the starting of input readers and output writers. If assignments were specified at system generation time, you do not have to respecify them, unless you wish to make a modification. All of your assignment requests are made through operator commands.

Your ability to switch input/output devices to online or offline status lets you modify their allocation. Online

devices can be assigned to problem programs, while offline devices cannot be. Your main reason for placing a device offline is to reserve it for preventive maintenance. To switch a device's status, enter a VARY command indicating the device and the desired status.

By issuing the MOUNT command, you can cause an input/output device to be assigned to those job steps that require the parti-

cular volume mounted on it. For example, you might give a MOUNT command for a device when you know that a volume will soon be used by many independent jobs.

In systems with the automatic volume recognition option, you can mount volumes on online devices that are not ready, thus anticipating the later needs of jobs you are scheduling.

Volume Mounting: In most installations, your role with respect to I/O devices is to mount and dismount volumes.

The job scheduler, using information from data definition statements, determines the input/output resources to be assigned to a job and the volumes that are required. If these volumes are not mounted, the job scheduler writes you a mounting message. Each message states that either a specific volume or a scratch volume is to be mounted. Mount the requested volume and press the START key on the device to continue processing.

You should never mount a blank tape volume. The system checks for a tape label and the absence of data causes the whole volume to be scanned for a data record. If an unlabeled tape is required, a tape mark should be written to avoid unnecessary scanning. For a description of a program you can use to create labels on standard label tapes, see the section on the program IEHINITT in the publication IBM System/360 Operating System: Utilities.

After you mount the volume and ready the drive, the system reads the volume label. If an incorrect volume is mounted, the system repeats the mounting message and unloads the incorrect volume, if possible (some devices, such as the 2311, can only be unloaded manually). If a request was made for a tape volume without a standard label and if the volume mounted does not have a standard label, that volume will be accepted. The volume is treated as unlabeled, or as a volume labeled with non-standard labels, according to the DD statement.

The following volume mounting options can be selected at system generation time:

- Imperative mount. Mounting messages are written when the job step requiring the volumes is started.
- Automatic volume recognition. You take the initiative and mount labeled volumes on any unused drives. The system recognizes and remembers these volumes and assigns the drives to later job steps.

If your system has the automatic volume recognition feature, mount volumes you want the system to find for the first job at IPL before issuing the START command. Also before issuing START, be sure that all offline devices are known to the system by using the VARY OFFLINE command. After the first job, you can mount ahead for several jobs at a time. In addition, the system may ask you to mount other volumes, and you can mount these on any appropriate online

devices that are not ready. (Do not unload any units - you can only mount on units unloaded by the system.)

Automatic volume recognition handles nine-track tape, seven-track tape, and 2311 and 2314 direct-access devices. The density for seven-track tape is set at 200, 556, or 800 bytes per inch at the time the system is generated.

When volumes are to be dismounted, the system unloads the devices, if possible, and writes you messages identifying the volumes being unloaded.

Make ready all devices needed by all the jobs in the system at one time.

2321 Note: When you receive a mounting message for a 2321 data cell, press the RESET button on the device if the requested cell is already positioned properly. If you have to open the door on the unit to position the cell, then you do not have to press RESET -- closing the door performs the same function in this case.

#### Allocation Guidelines

When the scheduler cannot satisfy requests for allocation from available (online) devices, it sends you a message and a list of offline devices and reserved devices. You then either cancel the job or make an offline device available by replying to the message with a three-character device name.

Use the MOUNT command to reserve a volume on a device, when you know that several jobs are going to need that volume.

Volumes reserved through a MOUNT command are not dismounted by the system until an UNLOAD command is given, causing the system to unload the volume.

Device Names: When referring to I/O devices in the unitname parameters of operator commands, you must use the unique unit names assigned to each device. Symbolic device names of one to eight alphanumeric characters may be defined by your installation, but these are for use by your programmers in their data definition statements. Do not use symbolic names in operator commands.

#### OPERATOR COMMANDS

This section contains a description of the commands you use to give control information to the operating system. The formats, functions, parameters, and options of

the commands are included. For convenience, the commands are presented in alphabetical order.

The commands usable in systems with MFT are:

CANCEL	SHIFT
DISPLAY	START
MOUNT	STOP
REPLY	UNLOAD
REQ	VARY
SET	

Console commands, other than SET, START, SHIFT, and REPLY, are accepted whenever you issue them. To issue a SET or START command, first issue a REQ command. A REQ command makes the system pause and issue a message requesting further commands at the end of a job step. Any commands issued in response to such a message must be followed by a START command with no parameters. This command gives control to the reader to start the next job step. A CANCEL command is executed when it is issued, even if it refers to a job in a partition other than the one the scheduler is in.

The following conventions are used in illustrating the format of commands:

- Required letters (those shown in upper case) must be entered, but can be entered in either upper or lower case.
- Lower-case letters indicate that a parameter must be substituted.
- Dotted lines ... (indicating a series of terms), brackets [ ], and braces { } are not entered.
- Entries within brackets [ ] are optional.
- Entries within braces { } are required - you must select one.
- Numbers and punctuation marks (other than dotted lines, brackets, and braces) must be entered as shown.

Command formats are essentially free form, but one or more blanks must follow the operation field. Commands cannot occupy more than one line. For example, if a command is entered through a card reader, it may not be more than 80 characters in length.

If comments on commands are necessary, they should appear to the right of the operand field and be separated from it by at least one blank. If the operand field is null, a comma followed by at least one blank indicates that comments will follow.

### CANCEL -- Terminate Job Immediately

The CANCEL command is used to immediately terminate the scheduling or execution of a job. Optionally, you may request that an abnormal-end-of-task storage dump be taken if the command is received while the job is running. This command is always executed as soon as it is received.

A canceled job is not terminated unless the scheduler is assigned to the partition in which the job is being executed. If the job to be canceled is in a partition of higher priority than the one to which the scheduler is assigned, the CANCEL command causes the job to be canceled but not terminated. Further scheduling cannot take place in that partition until jobs in lower priority partitions have been either terminated or canceled. To permit scheduling in that partition, you must use the CANCEL command and the SHIFT command.

Operation	Operand
CANCEL	jobname[,DUMP]

jobname

specifies the name of the job to be terminated. The maximum length of a job name is eight characters.

DUMP

specifies that an abnormal-end-of-task storage dump is to be taken if a step of the job is being executed when the command is received. If the programmer has put in the SYSABEND data definition statement, a full dump is taken. If he has not included this card, an indicative (partial) dump is taken.

**Note:** You may be asked by your programmers not to use the CANCEL command on certain jobs. These jobs alter data sets containing information vital to the system -- canceling the jobs might make the data unusable.

### DISPLAY -- Cause Console Display

The DISPLAY command is used primarily to cause a console display of the name of each job at the time it is initiated, and at the time it is terminated. (If a job is terminated due to unusual circumstances, you will receive a message even if you have not used the DISPLAY command.) The command provides you with job name information needed for effective use of the CANCEL command and, together with system messages, keeps you informed of which jobs are currently being executed.

Operation	Operand
DISPLAY	{JOBNAMES} {STATUS}

**JOBNAMES**

specifies that the name of each job is to be displayed both when the job starts and when it terminates, and that unit record allocation is to be displayed when the step starts.

**STATUS**

specifies that the data set names and volume serial numbers of data sets with dispositions of KEEP, CATLG, or UNCATLG, are to be displayed on the console at step termination and job termination.

**MOUNT -- Allocate Device**

The MOUNT command is used to allow the allocation of an input/output device to all jobs in the system that require a particular volume without regard to the partitions in which the jobs are being executed.

Operation	Operand
MOUNT	unitname [,VOL=(NL,serial)]  [ ,USE={STORAGE PUBLIC PRIVATE} ]

**unitname**

specifies the name of the input/output device to be allocated. Unitname must specify a device that has been unloaded by the system.

**VOL=(NL,serial)**

specifies that the volume does not have standard labels. The alphanumeric serial number, up to six characters long, is used for allocation references. This parameter is not used for direct-access volumes.

**USE=STORAGE or PUBLIC or PRIVATE**

specifies that a direct-access volume will be used as either a storage volume or a public volume or a private volume. If this operand is not used, the system treats the volume as a private volume. A storage volume is the most freely allocated kind of volume, open to use by the largest variety of data sets, temporary or non-temporary. Slightly restricted is

a public volume, which can be allocated freely for temporary data sets, but which must be specified by volume serial number to be allocated to non-temporary data sets. A private volume is the least freely allocated kind of volume -- it is allocated only if its volume serial number is specified.

**2321 Note:** When issuing this command for a 2321 data cell, unitname must contain a bin number in addition to the channel, control unit, and device. For example, 263/8 is the unitname for bin number 8 of a 2321 with a device number of 3, a control unit number of 6, and a channel number of 2.

**REPLY -- Reply to Information Request**

The REPLY command is used to reply to messages from the operating system and from problem programs that request information.

The REPLY command need not directly follow the message requesting the reply. The message id ensures that the message is routed by the system to the correct job. Because other messages can be printed before the reply has been entered, be sure that, in replying to the last message printed, previous messages are not ignored.

Operation	Operand
REPLY	id, 'text'

**id**

specifies the 2-character message identification field of the message requesting the reply. This field is described in Chapter 5.

**text**

specifies the text to be entered in response to a message. The information passed to the program expecting the reply does not include the enclosing apostrophes. When using the REPLY command to answer system messages, always be sure to use upper case letters in the text.

**REQ -- Request Commands**

The REQ command is used to cause the system to inform you when you can enter commands. This is indicated by a READY message from the master scheduler.

Other than at IPL-time, you must issue a REQ command, and wait for a READY message from the master scheduler, before entering a START or SET command.

The REQ command is used to reschedule a higher priority partition and resume normal job sequence in a lower-priority partition. This is accomplished by using the REQ command together with the CANCEL and SHIFT commands. The REQ command causes the scheduler to pause after the job step in the lowest-priority partition has been terminated and issue a READY message. Cancel intervening jobs, place the job for the higher-priority partition (preceded by a dummy JOB card) in the input stream, and enter the SHIFT command. The system reassigns the scheduler to the requested partition, and the job placed in the input stream is scheduled for execution in that partition.

Operation	Operand
REQ	

SET -- Set Date, Time, and Location

The SET command is used to establish the date, the time of day, the device for the input work queue and whether the queue is to be formatted, or the location of the procedure library. Any combination of these may be specified.

The SET command can be issued only after you have received a READY message. The first SET command after initial program loading must always include the DATE operand.

In systems that do not include the timer option, the CLOCK parameter is ignored.

Operation	Operand
SET	DATE=yy.ddd[,CLOCK=hh.mm.ss] [,Q=(unitname[,F])] [,PROC=unitname]

DATE=yy.ddd  
specifies the date in the following format:

```
yy.ddd
|  |
|  |----Days (001-366)
|-----Year (00-99)
```

CLOCK=hh.mm.ss  
specifies the time of day in the following format:

```
hh.mm.ss
|  |  |
|  |  |----Seconds(00-59)
|  |-----Minutes(00-59)
|-----Hours (00-23)
```

If the new clock setting implies a change of date, the new date must be explicitly stated using the DATE parameter.

Q=(unitname[,F])  
specifies the name of the direct-access device on which the volume containing the input work queue (SYS1.SYSJOBQE) is to reside (do not specify a 2321 data cell in this operand). Space on this volume for the input work queue must have already been allocated. If this parameter is not given, the system assumes that the queue is on the system residence volume. (This parameter is used only in the initial SET command issued immediately after IPL.) When the input work queue is to be used for the first time, a comma and an F must be added to the 3-character unitname. The F is a request to the system to format the queue prior to the first job initiation; it must be used during the first IPL, but need not be used at subsequent IPLs. If the F is used at subsequent IPLs, it will cause reformatting each time it is used. (See the discussion under the heading "Restarting the System.")

PROC=unitname  
specifies the name of the direct-access device on which the volume that contains the procedure library resides. If this parameter is not given, the system assumes that the procedure library is on the system residence volume. (This parameter is used only in the initial SET command issued immediately after IPL and should only specify a device that is ready.)

SHIFT -- Reassign Job Scheduler to Higher Priority Partition

The SHIFT command is used to reassign the job scheduler to a partition of higher priority than the one to which it is currently assigned. This command is accepted but not executed when it is issued.

When the job in the partition to which the scheduler is assigned has been terminated, the system notes that a shift has been requested, and reassigns the scheduler to the next higher partition. If the job in that partition is waiting to terminate, either because it has reached normal completion or because it has been canceled, it is terminated by the scheduler.

If the number specified in the SHIFT command is 1, the shift process is complete at this point and the next job in the input stream is scheduled for execution in the partition just terminated. If the number is greater than 1, the process is repeated until the request is satisfied.

Execution of the SHIFT command is not complete until all jobs have been terminated in partitions of lower priority than the one to which the scheduler is to be reassigned.

Rescheduling is not complete until the job currently being executed in that partition has been terminated and the next job in the input stream has been scheduled for execution in that partition.

If a higher priority partition is to be rescheduled but the next job in the input stream is not to be scheduled for execution in that partition, you can enter the REQ command, causing the scheduler to pause after it terminates the job in the partition to which it is assigned.

The READY message is then displayed, allowing you to place a job for the higher priority partition into the input stream ahead of the job that was there. The new job must be preceded by a dummy JOB card.

After setting up the input stream, enter a CANCEL command for each job in intervening partitions, and then enter the SHIFT command. Alternatively, the CANCEL commands may be omitted and the shift process is executed as each job reaches termination.

Operation	Operand
SHIFT	x

x specifies the number of times the scheduler must be shifted to reach the necessary partition.

Note 1:

The SHIFT command causes the scheduler to be reassigned only to partitions of higher priority than the one from which the shift takes place. (Reassignment of the scheduler to partitions of lower priority is initiated by WAITR macro instructions within the jobs.)

If the number specified in the SHIFT command is greater than that required to reassign the scheduler to the highest-priority partition, the shift process is ended when the scheduler is shifted to the highest-priority partition.

For example, if the scheduler were assigned to P3 and a SHIFT 5 command were entered, the scheduler would be reassigned to P0; no "wraparound" would take place.

Note 2:

If the input stream has come to an end of file (RDR CLOSED), a SHIFT command issued before the next job is entered into the input stream is not recognized in time to permit this job to be shifted into the higher-priority partition.

In this case, a dummy JOB card (that is, any JOB card that is convenient) should be inserted ahead of the JOB card for the new job. This will cause the dummy job to be scheduled for execution in the lower-priority partition, and the new job to be scheduled for execution into the higher-priority partition.

Note 3:

By placing the SHIFT command in the input stream, you can maintain relatively continuous scheduling by pairing jobs that have approximately the same execution time and that have non-conflicting input/output requirements. If the SHIFT command is inserted in the input stream, it should be positioned directly ahead of the JOB card for the job that is to be scheduled into the higher-priority partition.

Use the command DISPLAY JOBNAMES with the SHIFT command in the input stream. This causes the scheduler to notify you of the beginning and ending of jobs in different partitions, and helps you keep a log of executed jobs.

START -- Start System Process

The START command is used to start an input reader or output writer.

The START command can be issued only after you have received a READY message.

Any sequence of commands submitted in response to a READY message must be followed by a START command with no parameters.

If the output writer is associated with a tape unit, the volume mounted must have a standard label.

Operation	Operand
START	{RDR}, unitname, {WTR} [volumeserial], [parmvalue], [DSN=datasetname,] [FILESEQ=file sequence number]

RDR specifies that an input reader is to be started.

WTR specifies that an output writer is to be started.

unitname specifies the name of the unit record or magnetic tape input/output device associated with the input reader or output writer that is to be started.

volumeserial specifies the alphameric serial number, up to six characters long, of a magnetic tape volume. If this parameter is specified, label checking is performed.

parmvalue specifies either an up-to-eight-character name of a job in the input stream or a one-character alphameric output class name. A job name is used only with RDR: when starting an input reader, giving the job name causes forward spacing through the input stream until the named job is found. An output class name is used only with WTR: giving a class name indicates the class of output the writer is to handle; if no class is specified, the writer is assigned to class A.

DSN=datasetname specifies the name of the data set associated with the input reader or output writer. The maximum length of a data set name is 44 characters. If this parameter is not specified, the data set name SYSIN is assumed for the reader.

FILESEQ=file sequence number specifies the file sequence number, up to four digits long, of a data set on a magnetic tape volume. This parameter is optional and is used only with RDR when unitname designates a magnetic tape device.

STOP -- Stop System Process

The STOP command is used to stop a console display of job names or of data set names, or to stop an output writer.

Operation	Operand
STOP	{JOBNAMES WTR, unitname STATUS}

JOBNAMES specifies that a console display of the names of jobs, initiated by the JOBNAMES parameter of the DISPLAY command, is to be terminated. For more information about JOBNAMES, see the discussion of the DISPLAY command.

WTR, unitname specifies that the output writer using the named unit is to be stopped by the system. This operand will not stop a writer assigned to class A -- only a new START WTR to class A will cause the presently active one to stop.

STATUS specifies the discontinuance of a console display, at step termination and job termination of the names and volume serial numbers of data sets with dispositions of KEEP, CATLG, or UNCATLG.

UNLOAD -- Prepare Volume for Dismounting

This command is normally used to remove a volume previously mounted in response to a MOUNT command.

The UNLOAD command causes a volume on an input/output device to be prepared for dismounting. When the volume is ready to be dismounted, you will receive a message. (The message may not be received until the current job is completed.)

The UNLOAD command causes the volume on the specified device to be unavailable to all jobs in the system.

Operation	Operand
UNLOAD	unitname

unitname specifies the unit address of the input/output device to be prepared for dismounting.

2321 Note: When issuing this command for a 2321 data cell, unitname must contain a bin number in addition to the channel, control unit, and device. For example, 263/8 is the unitname for bin number 8 of a 2321 with a device number of 3, a control unit number of 6, and a channel number of 2.

VARY -- Vary Status of Device

The VARY command is used to place an input/output device (other than a communications line) into an online or offline status.

Operation	Operand
VARY	unitname {,ONLINE ,OFFLINE}

unitname

specifies the unit address name of the input/output device whose status is to be changed.

| ONLINE

specifies that the device is to be made available for allocation by the job scheduler to problem programs.

| OFFLINE

specifies that the device is to be removed from the recognition of the job scheduler, and that any further allocation of the device to problem programs is to be prevented. If the device is in use (allocated to a problem program or to an input reader or output writer), the status is not changed until all the current users have finished with the device. When the status is changed to offline, you will receive a message. A device can be removed from the offline status only by a subsequent VARY command.

2321 Note: To vary the status of an entire 2321 data cell, use its three-character unit address -- 263, for example. To vary the status of a particular 2321 bin, use its five-character unit address -- 263/8, for example, where the 8 is the number of the particular bin being addressed.

#### SUMMARY OF SPECIAL MFT OPERATING TECHNIQUES

This section covers special procedures that do not apply to all three versions of the operating system. After reading this section, skip to Chapter 4, which describes general techniques.

#### HOW TO START A SYSTEM WITH MFT

1. Set the LOAD UNIT switches to the channel, control unit, and device to address the system residence volume.
2. Press the LOAD key and wait for the LOAD light to go out. If it does not go out after a short period of time, make sure the unit is ready, and then press the LOAD key again.

3. Wait for a LIST PARTITION SIZES message and the WAIT light to go on. If the WAIT light goes on without a LIST PARTITION SIZES message, display the current PSW, and follow the directions for the error code you will find in the right half of the PSW. If the wait light goes on but there is no error code in the PSW, see if you have mounted and addressed the system residence volume correctly.

4. Enter REPLY id, 'YES' if the sizes are to be listed, or REPLY id, 'NO' if they are not. If NO is entered, skip to item 10 - the system will issue the READY message.

5. The system prints the sizes of the partitions as defined when the system was generated, plus the message INDICATE CHANGE, LIST, OR END. If the partitions are not to be redefined, skip to item 9.

6. To redefine the size of partitions, enter each new size in the form REPLY id, 'P<sub>n</sub>=x' (where n is the number of the partition, and x is a decimal number not over 8 digits long) followed by the signal end-of-block (EOB). If you wish to redefine more than one partition size with one REPLY command, separate the redefinition equations by one blank (no comma is used): REPLY id, 'P<sub>0</sub>=x<sub>1</sub> P<sub>1</sub>=x<sub>2</sub> P<sub>2</sub>=x<sub>3</sub> P<sub>3</sub>=x', for example. If the number of partitions is not to be redefined, skip to item 8.

7. To reduce the number of partitions below that set at the time the system was generated, enter REPLY id, 'P<sub>n</sub>=LAST' and the signal EOB.

8. If a list of the partitions and sizes is needed, enter REPLY id, 'LIST' and the signal EOB.

9. Enter REPLY id, 'END' and the signal EOB to terminate redefinition. The system will issue the READY message.

10. After receiving the READY message, enter a SET command specifying the date and any other parameters the installation requires.

11. Enter the DISPLAY command with the JOBNAMES parameter, and any required commands (such as START commands for input readers or output writers, if automatic START commands have not been specified by your installation).

12. Enter a START command with no parameters as your last command.

When the nucleus is initialized, you may be required by your installation to alter some optional features. For descriptions of these features and instructions on how to alter them, see "How to Specify System Parameters" in Chapter 4.

#### HOW TO SHIFT THE SCHEDULER

Shifting the scheduler to a higher-priority partition is done by entering a SHIFT command with the appropriate number as the operand. If the reassignment is to take place as quickly as possible, precede the SHIFT by a CANCEL for each lower-priority partition. If no CANCEL is entered, the shift process is not completed until all jobs in lower-priority partitions have reached normal completion.

#### HOW TO RUN A JOB THAT UPDATES SYSTEM DATA SETS

Do not run other jobs concurrently with a job that updates system data sets (SYS1.LINKLIB, SYS1.SVCLIB, and SYS1.PROCLIB). Run a job that updates system data sets as follows:

- Make sure no other jobs are in the system. (Shift the scheduler to the highest priority partition.)
- Place the job that is to update the system data sets in an input device.
- Start a reader to that device.
- Normal processing may be resumed after completing the job.

#### HOW TO PRINT A SYSOUT TAPE

Execute the program called IEFPRTP to print out a tape prepared as SYSOUT. Punch the following cards and schedule them as a job:

```
//xxx      JOB      xxx
//xxx      EXEC     PGM=IEFPRT
//PRINTAP DD      DSNAME=SYSOUT,           c
//          UNIT=tapename,                 c
//          DISP=OLD,                       c
//          VOLUME=(SER=(serial 1,         c
//          serial 2,...)),                 c
//          DCB=DEN=d
//PRINTER DD      UNIT=printname
```

The variables are:

xxx which is defined by your installation.

tapename identifies the tape drive or drives you wish to use for input.

serial 1, serial 2, etc. are the serial numbers of the tape volumes you wish printed out.

d is a one-digit number specifying the density of the tape(s) to be printed.

printname is a unit address. It must specify a printer. If only one printer is in the system and it is being used for system output, printname should not be used. Instead, the last card would be:

```
//PRINTER DD SYSOUT=A
```

IEFPRT prints the entire contents of the SYSOUT tape or tapes, stopping when the special end record is encountered. This record is the only record of its data set and contains END OF OUTPUT as its first 13 characters.

Note: The system may not be able to print the contents of the SYSOUT tape or tapes if your programmers have specified the density for their SYSOUT data sets in DD statements or in DCB macro instructions.

The job scheduler in a system with MVT (Figure 4) is similar to the job scheduler in systems with the primary control program (or with MFT). The major difference from an operating point of view is the addition of a queue manager and the ability to start multiple readers, writers, and initiators.

The queue manager controls the input and output work queues. The input queue is a list of jobs in the order in which they are to be processed internally. The output queue is a list of data sets in the order in which they are to be handled by the output writer. Instead of simply running one after the other, jobs in this system are rearranged and stored in a queue to run according to their priorities.

Later, an initiator takes each job from the input queue on a priority basis and starts it. When the job is finished, the initiator puts requests for system output in the output queue for later handling by an output writer.

Several readers can put job control records in the input queue while, at the same time, the initiators are starting job steps and the output writers are handling system output.

The number of input streams existing at one time depends on the number of START RDR commands you issue - one command starts one reader and one input stream, two commands start two readers and two input streams, and so on. Each reader reads data and control statements, places the information from them in proper format for later use by the initiator, and arranges the jobs in priority sequence in the input queue.

An initiator takes jobs from the input queue, assigns I/O devices for each data set, sends you I/O mounting messages, and introduces each job step to the supervisor as a separate "task" - a unit of work to be done at the same time as other units of work.

When the job step or job ends, the initiator removes it from the system and releases the no-longer-needed I/O devices. At the end of the job, the initiator completes information in the output queue for the output writer to handle. The initiator then goes on to the next job in the queue.

Like the reader, the number of initiators in the system depends on your commands

-- one initiator is started for each START INIT command. The degree of multijobbing being done is thus within your control -- you start many input streams or few, and start many job steps or few, depending on your installation's needs.

An output writer, activated by your command, START WTR, writes system output data sets, according to job priority and output class, on external devices such as printers and punches, depending on specifications given in commands or in job control language in an input stream. The number of output streams depends on the number of output writers you start.

#### STARTING, STOPPING, AND RESTARTING THE SYSTEM

System operations are controlled mainly through a console I/O device - the IBM 1052 Printer-KeyBoard. You give commands to the system, and receive messages from it, through the console.

There may be one console or two, depending on your installation. If you have two, one of them is called the primary console and the other is called the alternate console. Either console may be a composite console, made up of a card reader and a printer.

The primary console is active until you shift to the alternate by pressing the INTERRUPT key on the system control panel. Before shifting, give the VARY command to switch the alternate console offline to avoid conflicts with executing programs. For procedures to follow when a console malfunctions, see "How to Bypass a Console Malfunction" in Chapter 4.

To give the system a command through an IBM 1052 Printer-KeyBoard, first see if the PROCEED light is on. If it is on, type in your command without further preparation. If it is not on, press the REQUEST key, wait for the PROCEED light to turn on, and then type in your command. The total number of typed characters per command cannot exceed 126. To mark the end of a command, signal end of block (EOB) by pressing the alternate coding key and the numeric 5 key.

You can cancel a typed line by pressing the alternate coding key and the numeric 0 key.

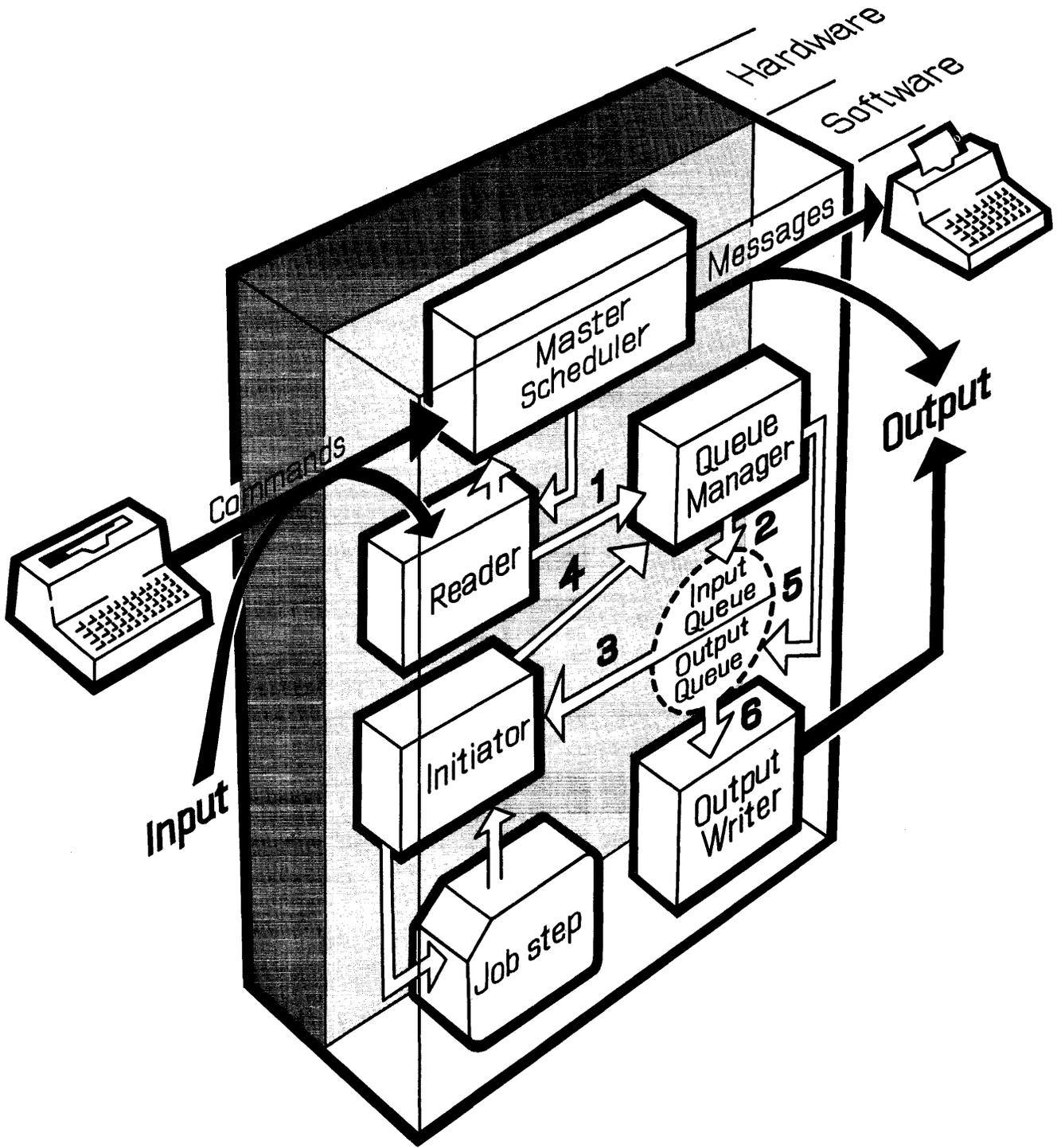


Figure 4. How the Scheduler Works in Systems With MVT

To give the system a command through a composite console (made up of a card reader and a printer, for example) press the card reader's STOP key, place the command in the reader, and press the START key and the EOF key.

You can also enter commands through a SYSIN (system input) stream by using command statements (see the topic "Controlling Jobs"). Each command statement, or group of command statements, must precede a JOB, an EXEC, or a null statement. Commands are accepted as soon as they are read - they are not synchronized with the jobs in the SYSIN stream.

#### STARTING THE SYSTEM

Starting the system includes initial program loading (IPL), readying the nucleus, and readying the scheduler.

#### Initial Program Loading

Initial program loading is a procedure carried out at the beginning of a shift, after a power-on following an electrical shutoff, after malfunctions that require reloading the control program into main storage, after scheduled maintenance, and as part of switching from one system to another.

Begin initial program loading by selecting the direct-access storage device on which the operating system resides: set the three LOAD UNIT switches on the control panel to the proper channel, control unit and device numbers; then press the LOAD key on the panel.

Pressing the LOAD key turns off the MANUAL light, turns on the LOAD light, and starts reading the IPL program from the input device. After the IPL program is read into lower main storage, control is passed to it, and the LOAD light turns off. If either the reading operation or the passing of control is unsuccessful, the CPU pauses and the LOAD light stays on.

When the IPL program gets control, it loads the nucleus of the control program into main storage. The IPL program loads a standard, or primary, nucleus unless you cause it to load a secondary nucleus. For a description of this procedure, see "How to Load a Secondary Nucleus" in Chapter 4. After the nucleus is loaded, control is given to a nucleus initialization program (NIP).

If the IPL program does not finish successfully, or if I/O errors occur while NIP is running, the WAIT light turns on and

an error code is placed in the low-order byte of the program status word (PSW). Whenever the WAIT light turns on without a message, display the PSW, note the error code, and follow the instructions for that code given in the publication IBM System/360 Operating System: Messages, Completion Codes, and Storage Dumps.

#### Readying the Nucleus

The nucleus initialization program (NIP) does general preparatory work for the system. If the communication option was specified at the time the system was generated, you will receive a message, SPECIFY SYSTEM PARAMETERS, requesting any changes.

If you receive this message, your system programmer may ask you to alter one or more options, such as the BLDL option, the RAM option, or the RSVC option. Explanations for the various options, and instructions on how to alter the options, are given in Chapter 4, under the heading "How to Specify System Parameters."

After NIP completes its preparation of the system, it passes control to the master scheduler. You will receive a READY message from the system and the WAIT light will turn on.

If an error other than an I/O error occurs during the running of NIP, the WAIT light turns on, and you will receive a message identifying the error. (No message is sent if the system console is not ready, but a code can be found in the right half of the current PSW as the system waits.)

#### Readying the Scheduler

After initial program loading, issue commands to start the job scheduler, which in turn begins the flow of work through the system.

When the system is ready to run, you will receive a READY message, and the WAIT light will go on. You may then enter commands.

Your first command must be a SET command specifying the date and the time of day. Optionally, SET can specify the names of the devices for the input queue and a procedure library (SYS1.PROCLIB), and can also specify input queue formatting. Normally, the formatting parameter (Q=(unitname,F)) can be left out after the first IPL, causing the scheduler to use the input queue as it was formatted earlier.

You will receive a SPECIFY JOB QUEUE PARAMETERS message after you issue the SET command with the formatting parameter (Q=(unitname,F)). Instructions telling how to answer this message are given under the heading "How to Specify Job Queue Parameters."

After issuing SET, you can enter any commands in any order. START RDR, START WTR, and START INIT must be issued if they have not been specified by your installation at system generation time. If your installation has already specified START commands, I/O devices are automatically allocated to an input reader and an output writer, and the commands are written out on the console as if you had keyed them in yourself.

If you want to override automatic START commands in systems with MVT, use the SET command with the AUTO operand.

Examples:

- 1. To start a system that has automatic START RDR, START WTR, and START INIT commands:

```
SET DATE=yy.ddd,CLOCK=hh.mm.ss,
  Q=(unitname,F)
```

- 2. To start a system that has automatic commands, but to suppress them, and to start a reader, a writer, and an initiator:

```
SET DATE=yy.ddd,CLOCK=hh.mm.ss,
  Q=(unitname,F),AUTO=NONE
START WTR,unitname
START RDR,unitname
START INIT
```

- 3. To start a system that does not have automatic commands, and to start a reader and two initiators, and to remove an I/O device from the system before processing:

```
SET DATE=yy.ddd,CLOCK=hh.mm.ss,
  Q=(unitname,F)
START RDR,unitname
START INIT
START INIT
VARY unitname,OFFLINE
```

To Improve Storage Use: The order in which you issue START commands can improve main storage utilization. Particularly on systems with 256K of main storage, start the longest-running tasks first, the shortest-running ones last. This practice will reduce the possibility of main storage fragmentation.

Number of Initiators to Start: The number of concurrent initiators to be started should be carefully regulated according to the needs of your installation. As a rule of thumb, if the wait light is on most of the time, you could probably start more initiators for greater performance. If the wait light is flickering rapidly, you are likely to have a good number of initiators going. If the wait light is out almost all the time, you may have too many initiators processing at the same time.

STOPPING THE SYSTEM

Issue a STOP INIT command for each initiator in the system, a CANCEL command (optional) for each job, a STOP RDR for each reader, a STOP WTR for each output writer, and a WRITELOG CLOSE command. Wait for all activity to cease after these commands, and then issue the HALT EOD command to move internal data from main storage to the SYS1.LOGREC data set. The system will send you a message when the data has been moved.

RESTARTING THE SYSTEM

Follow the same steps you took in starting the system, but leave out the formatting parameter, Q=(unitname,F), when you issue the SET command.

Make sure that all direct-access volumes containing system input data and system output data are ready. System input and system output volumes may be switched to a different unit of the same type than the one they were on when the system went down. If, however, the volumes were on a control unit having the record overflow feature, and the track overflow feature was used to record the data, then the volume that was on the control unit having the record overflow feature can only be switched to another control unit having the record overflow feature. All volumes with SYSIN must be mounted. All volumes with SYSOUT should be mounted. If it is not possible to mount a volume with SYSOUT and the message IEF421I INIT=jjj.sss(2) appears in the subsequent system restart, the jobs jjj must be placed on the HOLD Q until the volumes with SYSOUT can be mounted.

When a system output writer encounters a data set all of whose volumes are not mounted, the data set is bypassed. Message IEF304I will go to the console and output device and will tell you the data set name and the volume serials, which you can use to dump the data set at a later time.

The job scheduler will send you the names of any jobs and job steps being

handled by the reader or initiator before the failure, and the output writer will print out all the data on the system output data sets. You will receive messages like the following:

```
IEF420I RDR = jobname  
IEF421I INIT = jobname stepname (1 or 2)
```

The first kind of message indicates that the named job was being processed by the reader. Such a job can be reentered as is.

The second kind of message indicates that the named job step of the named job was somewhere between initiation and termination: A one (1) in the parentheses indicates the step was being processed by the initiator and had not yet started executing; a two (2) in the parentheses indicates that the step was executing or was being terminated by the initiator. A job named in this kind of message may require programmer analysis before it can be reentered.

For example, a job can be reentered without modification if it writes only on data sets with temporary or system-assigned data set names. But if it writes on data sets with non-temporary data set names, the programmer will have to make sure his data is still usable. If it is not, he may be able to alter his job control language statements to get around the unusable data before he resubmits his job.

## CONTROLLING INPUT AND OUTPUT

This section describes how to work with the operating system in controlling input and output, particularly in the area of allocating or assigning devices to jobs.

### INPUT

A system input (SYSIN) stream is made up of job control language (JCL) statements and problem program data routed by the control program to their destinations within the system. Sets of input data entering the system through the SYSIN stream are called SYSIN data sets. Input data can also be read directly by the problem program and not be a part of a SYSIN stream.

Data in the input stream is written on a direct-access volume for later reading by the problem program.

### Input Readers

An input reader is that part of the scheduler that reads a system input stream from a single device. You assign an input device to an input reader by issuing a START RDR command.

In systems with MVT, several input readers can bring jobs into the system from several SYSIN streams at the same time. In these systems, you not only issue a START RDR command to start a new input reader, but also can issue a STOP RDR command to end one. (Input readers stop automatically on encountering an end-of-file condition.) SYSIN data sets in an input stream are stored on non-demountable direct-access volumes while other programs are running. Job steps can later read the data sets at high speed from the direct-access volumes.

### OUTPUT

A system output (SYSOUT) stream consists of system messages and problem program output data sets routed by the control program to a common output device. Problem

program data sets that leave the system through the output stream are called SYSOUT data sets. Output data can also be written directly by a problem program and not be a part of a SYSOUT stream.

Problem programs write their output on a direct-access volume. The output writer later picks up the output for printing.

### Output Classes

Your installation, depending on its needs, can group SYSOUT data into as many as 36 different classes. Each class has a 1-character classname and includes all system messages and SYSOUT data sets the system associates with the classname. This permits the grouping of output data with a common characteristic. The characteristic might be a type of output device, a priority, or possibly a location - for example, the data for the third floor programmers might have a classname of 3.

Examples of possible classnames for other kinds of output are:

- A High priority printed output, such as messages from the control program.
- B High priority punched output, such as error records to be hand-corrected and rerun.
- C Low priority punched output, such as a summary dump of disk records for auditing.

The SYSOUT class for a set of system messages or a SYSOUT data set is specified in the JCL statements for the associated job. These statements are described in the manual IBM System/360 Operating System: Job Control Language.

The system messages for a job are in the class specified by the MSGCLASS parameter of the JOB statement. A SYSOUT data set is in the class specified by the SYSOUT parameter of the DD statement for the data set.

The JCL statements in the following example describe a one-step job that executes PROGRAM in a system with MVT. The system messages for this job are in class B, while the output data set is in class C.

```
//EXAMPLE JOB MSGCLASS=B
//STEP1 EXEC PGM=PROGRAM
//RESULT DD UNIT=2311,SYSOUT=C, c
// SPACE=(132,(1000,10))
```

## Output Writers

An output writer is that part of the scheduler that writes a single system output stream to a single device. You assign an output device to an output writer by issuing a START WTR command.

In systems with MVT, several output writers can run at the same time on several SYSOUT streams, and more than one writer can work on an output class. In these systems, you not only issue a START WTR command to start a new output writer, but also must issue a STOP WTR command to end one.

By using START WTR, you specify the association between classnames and devices, by naming (a) the device to be allocated to the output writer and (b) the classnames that the output writer is to process. You can thus ensure that each SYSOUT class is written on the device that is most suitable for that class. Given the example in the preceding section, you might issue the following command, assuming that 00E is a printer.

```
START WTR,00E,,C
```

This would cause the output data set to be printed. On the other hand, (assuming 185 is a magnetic tape drive) if you issued:

```
START WTR,185,,C
```

the data set would be written on magnetic tape. If you wanted a printed listing from this tape, you could use one of the tape SYSOUT programs described under the heading "How to Print a SYSOUT Tape."

A job step writes SYSOUT data sets at high speed onto a direct-access device. Later, after the job is done, an output writer will handle the data sets concurrently with other jobs (Figure 5).

## System Log

The system log is kept by the system on a permanently mounted direct-access volume. The system log is available for use by any program at your installation.

There are three kinds of information that can appear in the system log:

- Job time, step time, and data from the JOB and EXEC statements of a job that has ended. This information is entered on the log by an accounting routine written at your installation.
- Operating data entered by problem programs using a write to log (WTL) macro instruction.

- Descriptions of unusual events that occurred during your shift. You can enter these descriptions by issuing a LOG command.

If you want the system log printed out, you can issue a WRITELOG command.

Use the LOG command to make entries into the system log. When you want to output the information in the log, use the WRITELOG command with a message class specified. The system will close the data set being used and open the other one. You will get a "high expiration date" message naming one of two data sets -- SYS1.SYSVLOGX and SYS1.SYSVLOGY -- that together make up the system log. Make a note of the name of the data set and reply with a 'U' showing the system that it may use the data set. Keep track of which data set is being used -- that is, which one is open. When the system is being stopped, and you know that SYS1.SYSVLOGX is open, issue a WRITELOG CLOSE command to close the data set -- this data set is automatically opened first, at the point at which you closed it, when the system is restarted. If SYS1.SYSVLOGY is open when the system is being stopped, do not use WRITELOG CLOSE -- instead issue a WRITELOG command with a message class. This data set is written over when the system is restarted, and if you do not have it outputted, its information may be lost.

## ALLOCATING DEVICES

Device allocation is the assignment of I/O devices for use by job steps or the system. If a job step is using a tape drive, no other job step may use the tape drive until the first job step is finished -- the entire device is assigned. If the job step is writing on a direct-access volume, however, it may only be necessary to assign a particular set of tracks; data sets for many job steps can be on the same volume.

Device allocations are made in response to requests from three sources:

- Data definition statements. These may specify the I/O requirements of the job steps in the system.
- System generation statements. These may request fixed assignments for system processes, such as the automatic starting of an input reader.
- Operator commands. These may request assignments for system processes, or may modify assignments made when the system was generated.

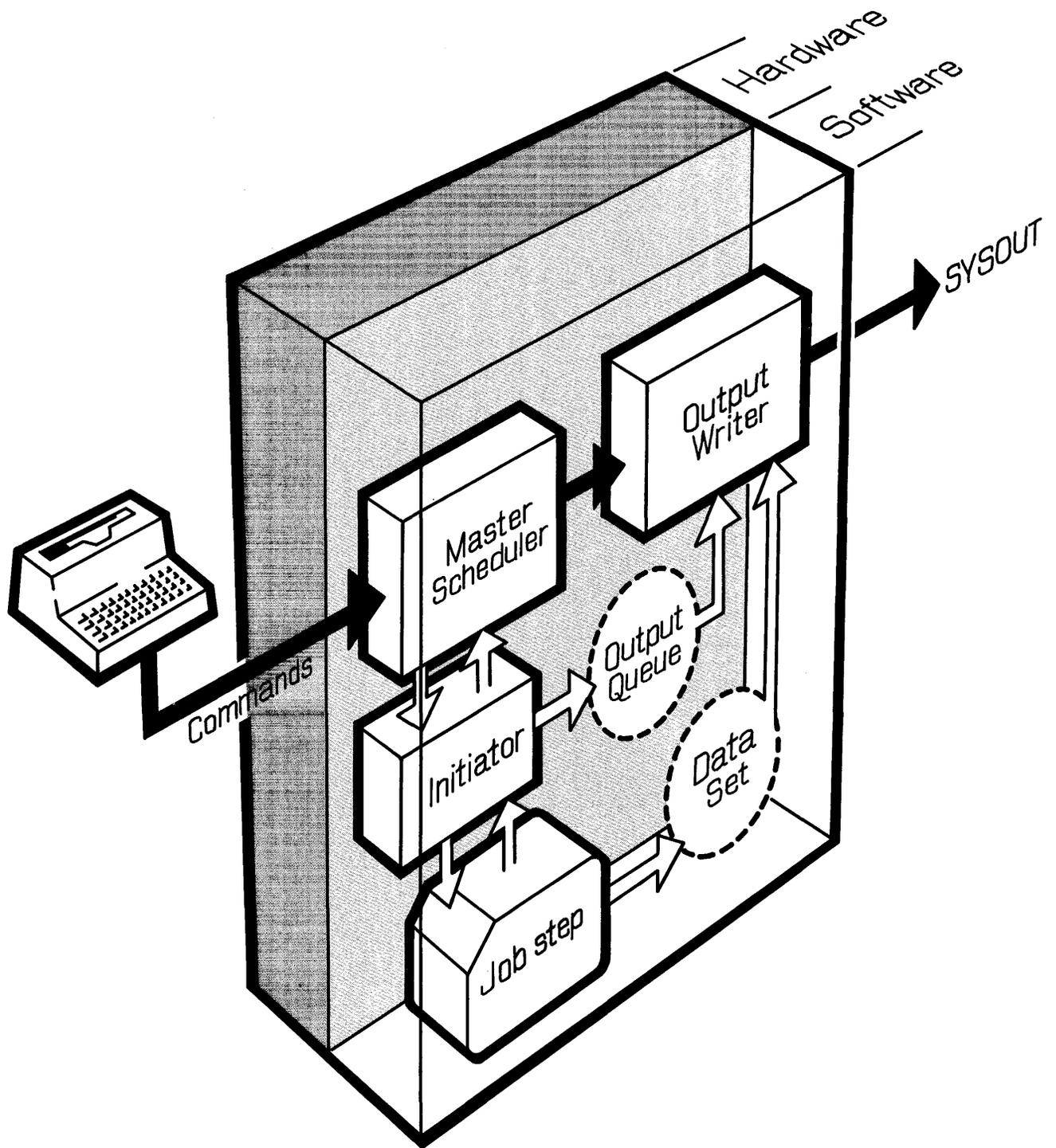


Figure 5. Output Process in Systems With MVT

### Assignment by the Scheduler

The job scheduler assigns I/O resources needed for the data sets of a job step. A programmer specifies these input/output requirements in his data definition statements. Using this information, the job scheduler attempts to provide overlapped operation, conserve input/output resources, and recognize items that increase input/output efficiency. The job scheduler then allocates I/O devices directly to the job step.

### Assignment by the Operator

You are responsible for device assignments made for the starting of input readers and output writers. If assignments were specified at system generation time, you do not have to respecify them, unless you wish to make a modification. All of your assignment requests are made through operator commands.

Your ability to switch input/output devices to online or offline status lets you modify their allocation. Online devices can be assigned to problem programs, while offline devices cannot be. Your main reason for placing a device offline is to reserve it for preventive or corrective maintenance. To switch a device's status, enter a VARY command indicating the device and the desired status.

By issuing the MOUNT command, you can cause an input/output device to be assigned to those job steps that require the particular volume mounted on it. For example, you might give a MOUNT command for a device when you know that a volume will soon be used by many independent jobs.

Volume Mounting: In most installations, your role with respect to I/O devices is to mount and dismount volumes.

The job scheduler, using information from data definition statements, determines the input/output resources to be assigned to a job and the volumes that are required. If these volumes are not mounted, the job scheduler writes you a mounting message. Each message states that either a specific volume or a scratch volume is to be mounted. Mount the requested volume and press the START key on the device to continue processing.

Occasionally you will receive two mount messages in a row for the same volume -- one message starting with IEF and the other

with IEC. Treat the two messages as though they were one. (The second is merely a reminder if you have not yet answered the first.)

You should never mount a blank tape volume. The system checks for a tape label and the absence of data causes the whole volume to be scanned for a data record. If an unlabeled tape is required, a tape mark should be written to avoid unnecessary scanning. For a description of a program you can use to create labels on standard label tapes, see the section on the program IEHINITT in the publication IBM System/360 Operating System: Utilities.

After you mount the volume and ready the drive, the system reads the volume label. If an incorrect volume is mounted, the system repeats the mounting message and unloads the incorrect volume, if possible (some devices, such as the 2311, can only be unloaded manually). If a request was made for a tape volume without a standard label and if the volume mounted does not have a standard label, that volume will be accepted. The volume is treated as unlabeled, or as a volume labeled with non-standard labels, according to the DD statement.

When volumes are to be dismounted, the system unloads the devices, if possible, and writes you messages identifying the volumes being unloaded.

2321 Note: When you receive a mounting message for a 2321 data cell, press the RESET button on the device if the requested cell is already positioned properly. If you have to open the door on the unit to position the cell, then you do not have to press RESET -- closing the door performs the same function in this case.

### Allocation Guidelines

When the scheduler in systems with the primary control program cannot satisfy requests for allocation from available (online) devices, it sends you a message and a list of offline devices and reserved devices. You then either cancel the job or make an offline device available by replying to the message with a three-character device name.

In systems with MVT, however, when the scheduler cannot allocate, it may be because of an additional factor: other jobs or the system itself may be using the device needed by the job being scheduled.

When the scheduler sends you a message because it cannot allocate, it lists any

offline devices as well as any devices assigned to the system, such as tapes or disks assigned to the control program's reader or writer routines and may suggest that you reply with WAIT, NOSEP or CANCEL.

When the scheduler has no other alternative but to wait for devices to become free, it puts itself in a wait state after sending you an IEF388I WAITING FOR DEVICES

message. The system waits for devices to become free unless you cancel the job while the system is waiting.

When the scheduler has no other alternative but to ignore the SEP option requested by the job control language, it attempts allocation recovery after sending you an IEF389I SEP REQUEST IGNORED message.

Depending on the message you receive, reply with the word 'CANCEL', with a unit name, with 'WAIT', or with 'NOSEP'. Replying with 'WAIT' tells the system to wait for devices to become free, while 'NOSEP' tells it to ignore job control language requests to provide channel and unit separation for the job step's data sets.

When answering this message, be sure to use the REPLY command and to spell your reply word in upper case letters only. If instead you use the CANCEL command, the system will not terminate the job even though you may get a message saying the job is canceled -- the system will wait for you to use the REPLY command. You can however use the CANCEL command later after using the REPLY command to say WAIT, if you change your mind about wanting the job to continue waiting.

If you ask the system to wait, you can issue a STOP RDR or STOP WTR command to free a device needed by the present job step. If you ask the system to wait, and later change your mind, you can still cancel the job. If you reply with NOSEP and this still does not help the scheduler to assign a device, you will get another message and another chance to tell the system to wait, cancel, or activate an offline device.

Use the MOUNT command to reserve a volume on a device, when you know that several jobs are going to need that volume.

Volumes reserved through a MOUNT command are not dismounted by the system until an UNLOAD command is given, causing the system to unload the volume.

Device Names: When referring to I/O devices in the unitname parameters of operator commands, you must use the unique unit names assigned to each device. Symbolic device names of one to eight alpha-numeric characters may be defined by your installation, but these are for use by your programmers in their data definition statements.

Do not use symbolic names in operator commands, except in the optional devicename parameter of the MOUNT and START commands. (Using devicename is better than using unitname - if you specified the unit name of a device the system was using, or was about to use, your command would be rejected.)

The number of devices associated with a symbolic name may range from one to the total number of devices in your installation. This allows the devices to be grouped according to whatever attributes

your installation considers significant - device type, for example, or special equipment, or installation configuration.

Work Volumes: Make sure there are sufficient work volumes available in the system to satisfy requests for temporary data sets at peak loads. Failure to do this can delay the allocation of a job step while it waits for direct access space to become free.

Where to Mount Active Volumes: At IPL, mount relatively full and active disk packs (for example, system packs) at the highest address locations available on the channel. Also, place system residence and system data sets (SYS1.MACLIB, for example) at the highest available address locations of different channels.

SYSOUT Writers with a Single Initiator: If device allocation has entered a wait for direct access space in a single initiator environment with no SYSOUT writer running, a SYSOUT writer cannot be started to free space. If this occurs, you may have to cancel the waiting job, depending on your installation's procedures. You can avoid this situation by starting a writer after IPL and having one always in the system.

Freeing Direct-Access Space: A wait for direct-access space will not always result in the allocation of a waiting data set. Even the eventual deletion of the temporary data sets on the candidate volumes still may not leave sufficient space on the volumes to satisfy the request. In addition if SYSOUT data sets in previous steps of the job (or previous steps of other currently active jobs) still occupy space that the data set requires, you may have to use the CANCEL command.

## OPERATOR COMMANDS

This section contains a description of the commands you use to give control information to the operating system. The formats, functions, parameters, and options of the commands are included. For convenience, the commands are presented in alphabetical order.

In systems with MVT, all commands can be entered at any time. For example, the VARY command can be used before the initial SET command.

In systems with MVT, you can use abbreviations as well as the full command name when keying in your commands. The usable names and abbreviations are:

CANCEL	C	REPLY	R
DISPLAY	D	RESET	E
HALT	Z	SET	T
HOLD	H	START	S
LOG	L	STOP	P
MODIFY	F	UNLOAD	U
MOUNT	M	VARY	V
RELEASE	A	WRITELOG	W

Be sure to use the correct abbreviations for operator commands. For example, use S for START and T for SET. If you inadvertently key in S for SET, the system assumes you are giving a START command, queues the command, and waits for a SET command.

The following conventions are used in illustrating the format of commands:

- Required letters (those shown in upper case) must be entered, but can be entered in either upper or lower case.
- Lower-case letters indicate that a parameter must be substituted.
- Dotted lines ... (indicating a series of terms), brackets [ ], and braces { } are not entered.
- Entries within brackets [ ] are optional.
- Entries within braces { } are required - you must select one.
- Numbers and punctuation marks (other than dotted lines, brackets, and braces) must be entered as shown.

Except for letters between apostrophes, lower case letters are translated to upper case before being handled by the command scheduler. Some commands require apostrophes in their operands. Be sure letters between apostrophes in these commands are upper case if they are meant to be processed as upper case. In addition, when using the REPLY command to answer system messages, always be sure to use upper case letters in the text between apostrophes.

Command formats are essentially free form, but one or more blanks must follow the operation field. Commands cannot occupy more than one line. For example, if a command is entered through a card reader, it may not be more than 80 characters in length.

If comments on commands are necessary, they should appear to the right of the operand field and be separated from it by at least one blank. If the operand field is null, a comma followed by at least one blank indicates that comments will follow.

Many operator commands need a region of main storage to execute in. Most of these commands require about 8K bytes of storage. But START and MOUNT need 52K. The system may reject these commands if enough main storage space is not available.

CANCEL -- Terminate Job Immediately

The CANCEL command is used to immediately terminate the scheduling or execution of a job. In addition, the command is used to stop the writing of an output data set currently being processed by an output writer. The command can also be used to cancel a system task in the device allocation process (you might wish to cancel such a task, for example, after receiving a mounting message that you cannot satisfy). Optionally, you may request that an abnormal-end-of-task storage dump be taken if the command is received while the job is running. This command is always executed as soon as it is received.

**Note:** If the initiator has issued any messages requiring a REPLY, you must REPLY to those messages to allow related jobs to be removed from the system.

If the named job is in the input work queue when the CANCEL command is issued, the job is marked canceled and scheduled for deletion from the system. If the job is on the hold queue, it is removed from that queue, marked canceled, and entered in the input work queue for later deletion.

To cancel a system task, use the CANCEL command, if the task is in allocation. The task is in allocation if an operator message with the prefix IEF appears. Therefore, if a system task needs operator intervention during allocation (such as mounting a pack), it can be canceled. The CANCEL command will not work after allocation is completed.

Operation	Operand
{ CANCEL }	{jobname [,DUMP]}
{ C }	{unitname }

jobname  
specifies the name of the job or system task to be terminated. The maximum length of a job name is eight characters.

DUMP  
specifies that an abnormal-end-of-task

storage dump is to be taken if a step of the job is being executed when the command is received. If the programmer has put in the SYSABEND data definition statement, a full dump is taken. There is no indicative dump facility in systems with MVT.



- a1 will be either the beginning address of the region, or zeros if SYSINIT is waiting for work, main storage or a data set.
- a2 will be either the end address of the region, or zeros if SYSINIT is waiting for work, main storage or a data set.
- qs will be the number of bytes, in decimal, of supervisor queue space required.

If j is SYSINIT, s is a job name, and region limits a1 and a2 are given, the initiator is initiating or terminating a step. If j is SYSINIT, s is a job name, but a1 and a2 are zeros, the initiator is waiting for main storage or a data set. If j is SYSINIT, s is blank, and region limits are zero, the initiator is waiting for work; if jobs are being held, you can release them with an appropriate form of the RELEASE command.

HALT -- Prepare for Power-off

The HALT command is used before you turn the power off at the end of the day, or anytime the computer is not to continue under the control of the operating system. You must use this command to ensure that important statistics and data records in main storage are not lost permanently. This command should be preceded by a WRITE-LOG CLOSE command.

Operation	Operand
{HALT}	EOD
{ Z }	

EOD specifies that end-of-day storing is to be done of internal I/O device error counts. The information is stored in the SYS1.LOGREC data set (see the topic "Hardware Debugging Aids" in Chapter 4).

When the storing is done, the system sends you a message EOD SUCCESSFUL. At this point, you can safely turn the power off.

HOLD -- Temporarily Suspend Job Selection

The HOLD command is used to temporarily prevent one job, or all jobs in the input work queue from being selected for processing. If the named job has already been selected, or if it is not in the input work

queue, you will receive a message. Jobs temporarily suspended by HOLD are subject to CANCEL and RESET commands.

The HOLD command works in two different ways, depending on whether you use the jobname operand or the Q operand. A HOLD jobname command causes the job to be withheld from initiation until a RELEASE jobname command is given. A HOLD Q command, on the other hand, prevents the selection of all jobs in the input work queue until a RELEASE Q command is given. (A RELEASE Q command will not release a job held by a HOLD jobname command, nor will a RELEASE jobname command release a job held because it was on the queue when a HOLD Q command was given.)

Operation	Operand
{HOLD}	{jobname}
{ H }	{ Q }

jobname

specifies the name of the job whose selection is to be suspended. The maximum length of a job name is eight characters. Although any job name can be in parentheses, a job with the name Q must have the Q in parentheses in the command statement.

Q

specifies that selection of all jobs from the input work queue is to be suspended.

LOG -- Store Information in Log

The LOG command is used to enter information into the system log.

Operation	Operand
{LOG}	'text'
{ L }	

text

specifies the exact text you wish to enter into the system log. The message written in the system log does not include the enclosing apostrophes.

MODIFY -- Alter Output Writer

The MODIFY command is used to change the characteristics of an operating output writer. You can change the output classes associated with the output writer, change the conditions under which the output writer pauses for servicing of its device, or both.

Whenever the output writer pauses, it writes a message requesting you to perform any necessary action on its device. If the pause results from a new form number specification, you are given the form number.

Operation	Operand
{ MODIFY F }	identifier [,CLASS=(classname,...)] [,PAUSE={FORMS DATASET}]

**identifier**  
specifies either the identifier you used in starting the procedure or, if you did not use an identifier, the devicename you specified.

**CLASS=(classname,...)**  
specifies one to eight single-character names of the classes to be associated with the output writer. If only one classname is specified, the parentheses are omitted. If more than one classname is specified, the writer treats the specified classes on a priority basis, where the left-most character indicates the highest-priority output class.

**PAUSE=FORMS**  
specifies that the output writer is to pause when a change in forms on its device is necessary.

**PAUSE=DATASET**  
specifies that the output writer is to pause before starting to process each data set.

**MOUNT -- Allocate Device**

The MOUNT command is used to allow allocation of an input/output device to all job steps that require a particular volume, without intervening dismountings and remountings of that volume.

Operation	Operand
{ MOUNT M }	{unitname } {,VOL=(NL,serial)} {devicename} {,VOL=(SL,serial)} [,USE={STORAGE PUBLIC PRIVATE}]

**unitname**  
specifies the name of the input/output device to be allocated. In systems with MVT, you can specify a loaded or an unloaded device. The system will

request that the correct volume be loaded if it is not already mounted.

**devicename**  
specifies the type of device to be allocated. After a device is allocated, you receive a mounting message.

**VOL=(NL,serial)**  
specifies that the volume does not have standard labels. The alphanumeric serial number, up to six characters long, is used for allocation references. This parameter is not used for direct-access volumes.

**VOL=(SL,serial)**  
specifies that the volume has standard labels. The alphanumeric serial number, up to six characters long, is used in label checking and for allocation references.

**USE= STORAGE or PUBLIC or PRIVATE**  
specifies that a direct-access volume will be used as either a storage volume or a public volume or a private volume. If this operand is not used, the system treats the volume as a private volume. A storage volume is the most freely allocated kind of volume, open to use by the largest variety of data sets, temporary or non-temporary. Slightly restricted is a public volume, which can be allocated freely for temporary data sets, but which must be specified by volume serial number to be allocated to non-temporary data sets. A private volume is the least freely allocated kind of volume -- it is allocated only if its volume serial number is specified.

**2321 Note:** When issuing this command for a 2321 data cell, unitname must contain a bin number in addition to the channel, control unit, and device. For example, 263/8 is the unitname for bin number 8 of a 2321 with a device number of 3, a control unit number of 6, and a channel number of 2.

**RELEASE -- Make Job Available for Selection**

The RELEASE command is used to resume job selection that has been suspended by the HOLD command. If the job is in the input queue in a canceled status, or if the job is not found, you will receive a message.

To release a specific job that was held through a HOLD jobname command, issue a RELEASE jobname command. To release all jobs held because they were on the input work queue when a HOLD Q command was given, issue a RELEASE Q command. (A RELEASE Q command will not release a job held by a

HOLD jobname command, nor will a RELEASE jobname command release a job held because it was on the queue when a HOLD Q command was given.)

Operation	Operand
{ RELEASE A }	{ jobname Q }

jobname specifies the name of the job to be made available for processing. The maximum length of a job name is eight characters. Although any job name can be in parentheses, a job with the name Q must have the Q in parentheses in the command statement.

Q specifies that all jobs in the input work queue are to be made available for processing.

REPLY -- Reply to Information Request

The REPLY command is used to reply to messages from the operating system and from problem programs that request information.

The REPLY command need not directly follow the message requesting the reply. The message id ensures that the message is routed by the system to the correct job. Because other messages can be printed before the reply has been entered, be sure that, in replying to the last message printed, previous messages are not ignored.

Operation	Operand
{ REPLY R }	id, 'text'

id specifies the 2-character message identification field of the message requesting the reply. This field is described in Chapter 5.

text specifies the text to be entered in response to a message. The information passed to the program expecting the reply does not include the enclosing apostrophes. When using the REPLY command to answer system messages, always be sure to use upper case letters in the text.

RESET -- Change Priority of Job

The RESET command is used to change the selection priority of a job in the input work queue. If the job has already been selected for processing when the command is issued, the priority is not changed, and you will receive a message. You will also receive a message if the job is not found in the input queue.

Operation	Operand
{ RESET E }	jobname, value

jobname specifies the name of the job whose priority of selection is to be changed. The maximum length of a job name is eight characters.

value specifies the value to which the job's priority is to be set. The value is a two-digit numeric field that may range from a low of 00 to a high of 14.

SET -- Set Date, Time, and Location

The SET command is used to establish the date, the time of day, the device for the input work queue and whether the queue is to be formatted, the location of the procedure library, or the automatic commands you wish to override. Any combination of these may be specified.

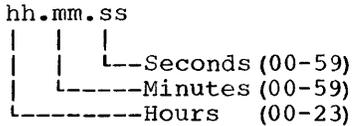
Always use the CLOCK operand in systems with MVT. These systems use the data in this operand when they name system data sets. (If you do not use the CLOCK operand, the system sets the clock value to zero.)

Operation	Operand
{ SET T }	DATE=yy.ddd, CLOCK=hh.mm.ss [, Q=(unitname [, F])] [, PROC=unitname] [, AUTO=characters]

DATE=yy.ddd specifies the date in the following format:

```
yy.ddd
  |
  |----Days (001-366)
  |
  |----Year (00-99)
```

CLOCK=hh.mm.ss  
 specifies the time of day in the following format:



If the new clock setting implies a change of date, the new date must be explicitly stated using the DATE parameter.

Q=(unitname[,F])  
 specifies the name of the direct-access device on which the volume containing the input work queue (SYS1.SYSJOBQE) is to reside (do not specify a 2321 data cell in this operand). Space on this volume for the input work queue must have already been allocated. If this parameter is not given, the system assumes that the queue is on the system residence volume. (This parameter is used only in the initial SET command issued immediately after IPL.) When the input work queue is to be used for the first time, a comma and an F must be added to the 3-character unitname. The F is a request to the system to format the job queue data set prior to the first job initiation; it must be used during the first IPL, but need not be used at subsequent IPLs. If the F is used at subsequent IPLs, it will cause reformatting each time it is used. (See the discussion under the heading "Restarting the System.") If the job queue data set is on the system residence volume, the formatting operand can be shortened to Q=(,F).

PROC=unitname  
 specifies the name of the direct-access device on which the volume that contains the procedure library resides. If this parameter is not given, the system assumes that the procedure library is on the system residence volume. (This parameter is used only in the initial SET command issued immediately after IPL and should only specify a device that is ready.)

AUTO=characters  
 specifies, in systems with automatic START commands, whether you wish to retain any of those commands. For each automatic command printed out by the system, follow the equals sign by a Y if you want to retain the command, or by an N if you want to override the

command. For example, if the system prints out S WTR, S RDR, S INIT, and you want to retain the automatic reader and writer but not the initiator, key in AUTO=YYN. If you want to reject all automatic commands, key in AUTO=NONE.

**START -- Start System Process**

The START command is used to start initiators as well as readers and writers.

Operation	Operand
{START}	{INIT
{S}	{procname}
	{(identifier)}
	{[,devicename]}
	{[,volumeserial]}
	{[,parmvalue]}
	{[,keyword=option,...]}

INIT  
 specifies that an initiator is to be started. No other operands are permitted when this operand is used.

procname  
 specifies the name of the procedure to be started. This name can be either a standard name or a special one provided by your system programmer. Standard procnames include RDR for an input reader and WTR for an output writer.

identifier  
 specifies an optional eight-character name you can add to identify a particular procname when you have started more than one. Each identifier you add must be unique when a different procname is specified. For example, if you started two readers named RDR, you could identify them by giving the first as S RDR(ONE) and the second as S RDR(TWO). You must not then issue S WTR(ONE) or S WTR(TWO) since a unique identifier is required. Later you could stop the one you wanted to stop by using the identifier -- P TWO, for example, would stop the second reader. In addition, if you do not specify an identifier but do specify a devicename, such as 00C, the device name becomes the identifier. You can then stop the task by issuing a STOP command with the devicename. For example, if you issued S RDR, 00C, you could later issue P 00C to stop the reader using the device with the name 00C.

**devicename**

specifies the name of an input or output device. This can be either a unit name (such as 280) or a general name (such as 2400). If specified, the name will override any corresponding unit specification in the cataloged procedure.

**volumeserial**

specifies the six-character serial number of a magnetic tape or direct-access volume. If specified, this parameter will override any corresponding volume serial specification in the cataloged procedure.

**parmvalue**

specifies parameter values to be passed to the program receiving control as a result of the START command. If more than one value is specified, parmvalue must be enclosed in parentheses, with commas separating the values. If one value is specified, no parentheses are needed. If parmvalue contains any non-alphameric character, such as an equals sign, parmvalue must be enclosed in parentheses. The two most common parmvalue types are the name of a job in the input stream and one or more system output class names. A job-name parmvalue is used only when starting an input reader: use this parmvalue to cause forward spacing through the input stream until the named job is met. Class-name parmvalues are used only when starting an output writer: use these parmvalues to limit the system output classes the writer is to process. The output writer treats the specified classes on a priority basis where the left-most character indicates the highest priority output class. If one or more class names are specified, all class names specified in the standard cataloged procedure are overridden.

**keyword=option**

specifies any appropriate keyword syntax allowable on a DD statement. (For detailed information on these keywords, refer to IBM System/360 Operating System: Job Control Language.) If such keyword parameters are specified, they will override the corresponding parameters on the DD statement for the input or output device in the cataloged procedure. If the devicename positional parameter is used, the UNIT keyword may not be used. If the volumeserial positional parameter is used, the VOLUME keyword

may not be used. If the input device is a disk, you must use the keyword DSN= to specify the correct data set. If the data set is not cataloged, you must use either the volumeserial parameter of the START command or the keyword VOLUME=SER=volume id. But whether or not the data set is cataloged, you must specify DISP=OLD when using the IBM-supplied reader procedures, unless you want the data set to be deleted.

**Note:** To make clear to the system which operands you are specifying when you are not specifying them all, insert commas for the missing operands in the middle of the string. For example,

```
START WTR,185,,C
```

lets the system know that you are not specifying the volumeserial operand.

**STOP -- Stop System Process**

The STOP command is generally used to stop the operation of an input reader, an output writer, or an initiator, or to stop a console display of job names.

A STOP RDR command will stop a reader after it has processed all input stream data for the job it was reading when you issued the command. After a reader is started, it must process one job before a STOP RDR command will take effect.

Operation	Operand
{STOP P}	{INIT procname,identifier identifier JOENAMES STATUS}

**INIT**

specifies that an initiator is to stop after initiating the current job. The system may be tied up under certain circumstances when the initiator is waiting for work and no work is coming into the job queue. For example, assume there are three jobs in the job queue and three initiators have been started. Also assume that the first job selected requires a region so large that neither of the two remaining initiators can get another region. When the first job terminates, the initiator does not release the region but rather goes to the job queue to select another job. At this point, if the job queue is empty, the entire system will wait, with a message, until you stop the initiator.

**procname,identifier**  
 specifies the name of the procedure started with a START command, followed by an identifier -- either one that you actually specified or, if you did not specify one, then the devicename that you specified. For example, if you specified S RDR(ONE), you could stop that reader by P RDR,ONE. If you specified S RDR,00C, you could stop that reader with P RDR,00C.

**identifier**  
 specifies either the identifier you specified when you started the procedure or, if you didn't specify an identifier the devicename you specified. For example, if you specified S WTR(JOE) you could stop the writer with P JOE. If you used S WTR,180 you could stop the writer with P 180.

**JOBNAMES**  
 specifies that a console display of the names of jobs, initiated by the JOBNAMES parameter of the DISPLAY command, is to be terminated. For more information about JOBNAMES, see the discussion of the DISPLAY command.

**STATUS**  
 specifies the discontinuance of a console display, at step termination and job termination, of the names and volume serial numbers of data sets with dispositions of KEEP, CATLG, or UNCATLG.

**Note:** When you use this command to stop a reader, a writer, or an initiator, the procedure does not stop immediately. Instead the system begins the stopping after the procedure finishes handling its current task. In addition, input readers-stop automatically on meeting an end-of-file condition.

UNLOAD -- Prepare Volume for Dismounting

This command is normally used to remove a volume previously mounted in response to a MOUNT command.

The UNLOAD command causes a volume on an input/output device to be prepared for dismounting. When the volume is ready to be dismounted, you will receive a message. (The message may not be received until the current job is completed.)

Operation	Operand
{ UNLOAD }	unitname
{ U }	

**unitname**  
 specifies the unit address of the input/output device to be prepared for dismounting.

**2321 Note:** When issuing this command for a 2321 data cell, unitname must contain a bin number in addition to the channel, control unit, and device. For example, 263/8 is the unitname for bin number 8 of a 2321 with a device number of 3, a control unit number of 6, and a channel number of 2.

VARY -- Vary Status of Device

The VARY command is used to place an input/output device (other than a communications line) into an online or offline status.

Operation	Operand
{ VARY }	unitname { ,ONLINE }
{ V }	{ ,OFFLINE }

**unitname**  
 specifies the unit address of the input/output device whose status is to be changed.

**ONLINE**  
 specifies that the device is to be made available for allocation by the job scheduler to problem programs.

**OFFLINE**  
 specifies that the device is to be removed from the recognition of the job scheduler, and that any further allocation of the device to problem programs is to be prevented. If the device is in use (allocated to a problem program or to an input reader or output writer), the status is not changed until all the current users have finished with the device. When the status is changed to offline, you will receive a message. A device can be removed from the offline status only by a subsequent VARY command.

**2321 Note:** To vary the status of an entire 2321 data cell, use its three-character unit address -- 263, for example. To vary the status of a particular 2321 bin, use its five-character unit address -- 263/8, for example, where the 8 is the number of the particular bin being addressed.

WRITELOG -- Write Out System Log

The WRITELOG command is used in systems with MVT to cause the contents of the system log to be written by a system output writer.

Operation	Operand
{ WRITELOG }	{ classname }
{ W }	{ CLOSE }

**classname**  
specifies the name of the system output class with which the contents of the system log are to be associated.

**CLOSE**  
specifies that the status of the system log is to be preserved. This operand must be used before stopping the system or data will be lost.

SUMMARY OF SPECIAL MVT OPERATING TECHNIQUES

This section covers information that does not apply to all three versions of the operating system. After reading this section, read Chapter 4, which describes general techniques.

HOW TO START A SYSTEM WITH MVT

- Set the LOAD UNIT switches to the channel, control unit, and device to address the system residence volume.
- Press the LOAD key and wait for the LOAD light to go out. If it does not go out after a short period of time, make sure the unit is ready, and then press the LOAD key again.
- Wait for a READY message and the WAIT light to go on. If the WAIT light goes on without a READY message, display the current PSW, and follow the directions for the error code you will find in the right half of the PSW. If the wait light goes on but there is no error code in the PSW, see if you have mounted and addressed the system residence volume correctly.

When the nucleus is initialized, you may be required by your installation to alter some optional features. For descriptions

of these features and instructions on how to alter them, see the topic "How to Specify System Parameters" in the next chapter.

After the READY message, the system will display any automatic commands selected at the time the system was generated, and wait for your SET command. You will be able to suppress any of the automatic commands in the system by using the special SET command operand, AUTO=characters. For a description of this operand and instructions on how to use it, see "SET." On receiving your SET command, the system will execute and display any automatic commands you did not suppress. Further commands can now be entered manually.

HOW TO SPECIFY JOB QUEUE PARAMETERS

You will receive the following message after you issue the SET command with the formatting parameter Q=(unitname,F):

IEF423A SPECIFY JOB QUEUE PARAMETERS

If your system programmer tells you not to alter the parameters that were set at the time the system was generated, issue REPLY id, 'U'. If he asks you to change the parameters, issue REPLY id, 'n,t,k' where:

- n is a number ranging from 10 to 255 specifying the number of queue records per logical track.
- t is a four-digit number specifying the number of queue records to be reserved for each initiator.
- k is a four-digit number specifying the number of queue records to be reserved for the termination of jobs that exceed t while being processed by an initiator.

The parameters are positional, and only those standing for parameters actually being changed need be given. For example:

REPLY id, '12,60,12' indicates a change to all three parameters.

REPLY id, '12' indicates a change to n only.

REPLY id, ',,5' indicates a change to k only.

## HOW TO DETERMINE SYSTEM STATUS

The DISPLAY ACTIVE (d a) command is your primary means of knowing what the system is doing. Use it frequently. Be alert to identical entries occurring on successive uses of d a. These entries indicate resource contention problems, such as contention for:

devices (check mount messages)

main storage (probable problem if proper volumes are mounted)

data sets (initiators may be waiting for data sets or main storage -- but they take up no region space)

You may need to determine system status more precisely when the system enters an enabled wait state or when one or more identical d a entries keep recurring. The following sequence of commands is a guideline for this.

HOLD Q	(prevents initiation of new jobs)
DISPLAY JOBNAMES	(get continuing data on job starting and stopping)
DISPLAY ACTIVE	(get current status)
STOP INIT	(one or more; if system appears to resume activity, wait for next job to complete processing before resuming normal initiator activity)
STOP all initiators	(this will eventually force one job waiting for a major portion of main storage to run -- stopping all but one initiator assures prompt execution of a big high-priority job)

CANCEL jobs (allows STOP INIT to be effective; should be done only as a last resort)

DISPLAY ACTIVE (should be used repeatedly to trace changing environment)

STOP RDRs and WTRS (to assure no main storage fragmentation)

This sequence can be terminated whenever the situation appears to be corrected.

Reduced system activity can also be due to the following:

- Exhausted system direct-access space. Readers that do not read to end-of-file may have been delayed waiting for space. Also, jobs that are waiting for allocation frequently cannot be assigned SYSOUT space. Both cases are normal, and will persist until space is freed as a result of printing and punching activity.
- Unsatisfied mount requests. Proper volumes should be mounted promptly, or the job canceled.
- When the system enters an enabled wait state and you have not held the job queue, the system is out of work if the number of IEF429I INPUT QUEUE EMPTY messages issued since the last IEF161I READER CLOSED ddd message is equal to the number of times you gave the START INIT command, provided no readers are currently active.
- Space in the system input queue has been exhausted. This can occur if the jobs being executed cannot complete because they are waiting for main storage, as indicated by DISPLAY ACTIVE output, and the readers are allowed to continue. The mechanism is: the readers continue filling the queue with jobs, the initiators cannot free queue space by starting jobs because there is not enough main storage, and the writers cannot free queue space by processing output because no jobs are progressing to completion.

If in response to your DISPLAY ACTIVE command you get a list with more than one master scheduler task on it, only the first is actually the master task. The others are system tasks such as altering the job queue or creating readers, writers, and initiators.

## HOW TO CONTROL JOBS THROUGH HOLD AND RELEASE

You can use HOLD and RELEASE to control scheduling of jobs in a multijobbing environment (more than one initiator). For example, assume that production jobs (setup) are multijobbed with one or more scientific (CLG) jobs. However, only one production job is running at a time. All production jobs can be initially put on the hold queue either through a TYPRUN=HOLD parameter on a JOB card or through a HOLD command after a reader has processed the jobs but before any initiators are started. The jobs can then be released by name through a RELEASE command one at a time -- when the first production job is complete the next can be released. DISPLAY JOBNAMES may be helpful in telling when jobs end, but since production jobs usually end in a predictable manner -- tapes rewind, keep messages are received, and so on -- it may not be needed.

The HOLD command remains in effect across system restarts -- that is, when the job queue is not reformatted. Jobs that were held individually or as part of the queue are still held after the system is restarted, and they must be freed by the appropriate form of the RELEASE command.

## HOW TO EXTRACT A JOB FROM TAPE INPUT STREAM

To extract one particular job from a tape input stream, enter:

```
S RDR,unit,,jobname
and then:
P RDR,unit
```

before readying the tape. Jobs preceding "jobname" will be skipped. "Jobname" is entered into the queues. Then the STOP is processed.

## HOW TO RUN JOBS THAT UPDATE SYSTEM DATA SETS

Do not run other jobs concurrently with jobs that update system data sets (SYS1.LINKLIB, SYS1.SVCLIB, and SYS1.PROCLIB); for example, do not run jobs that may try to use a cataloged procedure with a job that is updating that procedure. Run jobs that update system data sets as follows:

- Make sure no other jobs are active in the system; use the HOLD command to prevent the initiation of jobs on the job queue.
- Stop all readers, writers, and initiators.

- Place the jobs that are to update the system data sets in an input device.
- Start a reader to that device.
- Start one initiator when the reader stops after processing all the input.
- Start normal processing -- that is, start required readers, writers, and initiators -- when the first initiator stops after completing all the jobs. Use the RELEASE command to free the jobs placed on the hold queue earlier.

#### HOW TO PRINT A SYSOUT TAPE

In systems with MVT, use the utility program called IEBGENER to print out a SYSOUT tape. The IEBGENER program is described in the publication IBM System/360 Operating System: Utilities.

For example, you could punch the following cards and schedule them as a job (the x's are defined by your installation):

```
//xxx      JOB      xxx
//xxx      EXEC    PGM=IEBGENER
//SYSPRINT DD      SYSOUT=A
//SYSUT1   DD      DSN=SYSOUT,UNIT=2400, c
//          DISP=(OLD,DELETE),          c
//          VOLUME=SER=(VOL1,...,VOLn)
//SYSUT2   DD      UNIT=1403,           c
//          DCB=(LRECL=133,           c
//          BLKSIZE=133,RECFM=FM)
//SYSIN    DD      DUMMY
/*
```

The volume serial numbers specified in SYSUT1 reflect those produced in one pass of the SYSOUT writer, and must be specified in the same order as they were created.

#### STANDARD IBM MOUNTABLE-DEVICE NAMES

The following names are available for use in the devicename parameter of the MOUNT command. The names that you can use are defined by your installation at the time the system is generated.

- 2311 (any 2311 disk storage drive)
- 2400 (any 2400 nine-track magnetic tape drive with only 800 byte-per-inch density)
- 2400-1 (any 2400 magnetic tape drive with seven-track compatibility and without data conversion)
- 2400-2 (any 2400 magnetic tape drive with seven-track compatibility and data conversion)

- 2400-3 (any 2400 nine-track magnetic tape drive with only 1600 byte-per-inch density)
- 2400-4 (any 2400 nine-track magnetic tape drive with 800 and 1600 byte-per-inch density)
- 2314 (any of the 8 online drives of a 2314 direct-access storage facility)
- 2321 (any bin of a 2321 data cell)

#### START RDR AND WTR DEVICE NAMES

In START RDR or WTR commands, use any generic device names supported by QSAM, except direct-access device names. Supported names that can be used include:

Name	RDR	WTR
1403		x
1442	x	x
1443		x
2400	x	x
2400-1	x	x
2400-2	x	x
2400-3	x	x
2400-4	x	x
2501	x	
2520	x	x
2540	x	x
2540-2	x	x

#### UNIVERSAL CHARACTER SET (UCS) PROCEDURES

The UCS printer may be very effectively used in MVT by classifying the system output by the type of chain required. Then after loading a particular chain, you can start a system output writer to process the class of output corresponding to the chain.

The normal procedure used in PCP is not recommended for use in MVT. This consisted of executing the UCS utility as a step prior to the step which is to use the printer.

In MVT it is possible that a step of another job will be allocated the printer after the UCS utility has loaded the buffer in the first job. Therefore, the running of such jobs concurrently in MVT is not feasible. Control the mix so that no two jobs which load the UCS buffer run together. In addition, no SYSOUT writer may use a UCS printer unless you have loaded the buffer appropriately. Here again, make sure that no jobs using the UCS printer intervene between loading the buffer and starting the writer.

This chapter is a grouping of techniques and related data you will need to use in running the operating system. You will probably learn these techniques thoroughly, making revisions to them to suit your own needs. And of course you will be starting a "run book" of your own, adding tips and procedures as you discover them.

Before going over the techniques, this chapter reviews the purpose of some of the control panel keys, switches, and lights. (For further information about your particular model's control panel, console, or input/output devices see your hardware manuals.)

LOAD key	Loads an IPL program.	RATE switch	Sets the rate the CPU will operate at (used only when the CPU is already in stopped mode): PROCESS rate -- normal speed; INSTRUCTION STEP rate -- one whole instruction per push of the START key; SINGLE CYCLE rate -- one whole microprogram instruction per push of the START key.
START key	Starts instruction execution. (Works only if the CPU is in the stopped mode.)	STORAGE SELECT switch	Works only if the CPU is in the stopped mode. Selects the storage area addressed by the address switches:  FP -- floating point registers GP -- general purpose registers PSW -- current program status word MS -- main storage LS -- local storage SP -- storage protection
STOP key	Puts the CPU in the stopped mode.	LOAD light	Goes on when the LOAD key is pushed, goes off when the IPL record is loaded.
SYSTEM RESET key	Stops instruction processing, and resets the CPU, channels, online non-shared control units, and I/O devices. Does <u>not</u> reset registers.	WAIT light	Goes on when the CPU is in the wait state.
CHECK RESET key	Resets the console check lights.	MANUAL light	Goes on when the CPU is in the stopped mode.
INTERRUPT key	Stops program execution by causing an external interruption. Allows you to switch from the primary console to the alternate console.	TEST light	Goes on when the switches with a NORMAL position are not pointing to that position.
DISPLAY key	Displays information you specify by setting appropriate panel switches. Works only if the CPU is in the stopped mode.	SYSTEM light	Goes on when the CPU or channels are running.
ADDRESS COMPARE switch	Stops the CPU when it reaches any address you select in advance. Used to load a secondary nucleus at IPL time.	STORAGE DATA lights	Go on after the DISPLAY key is pressed when their corresponding keys are set to specify data to be loaded into storage or into registers.
LOAD UNIT switches	Tell the system where to get the IPL program when you push the LOAD key.	INSTRUCTION COUNTER/STORAGE ADDRESS lights	Go on after the DISPLAY key is pressed when their corresponding keys are set to specify the storage address where the CPU has stopped.

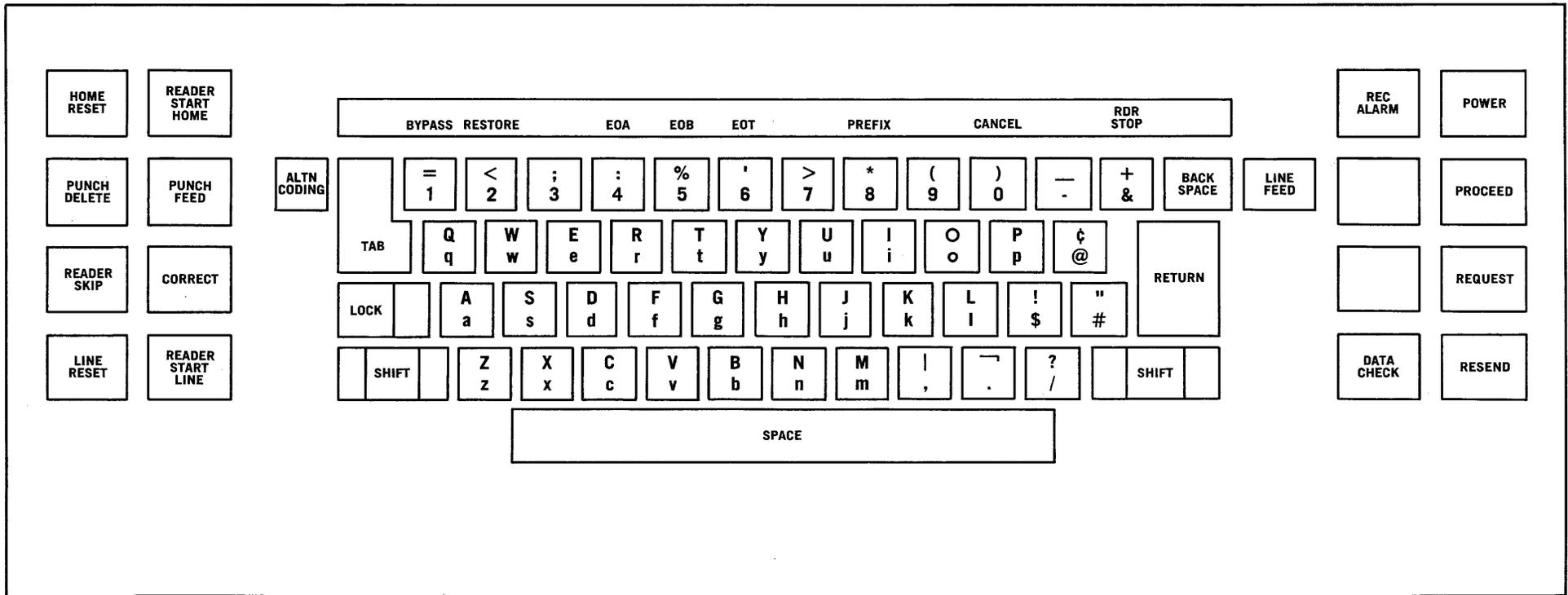
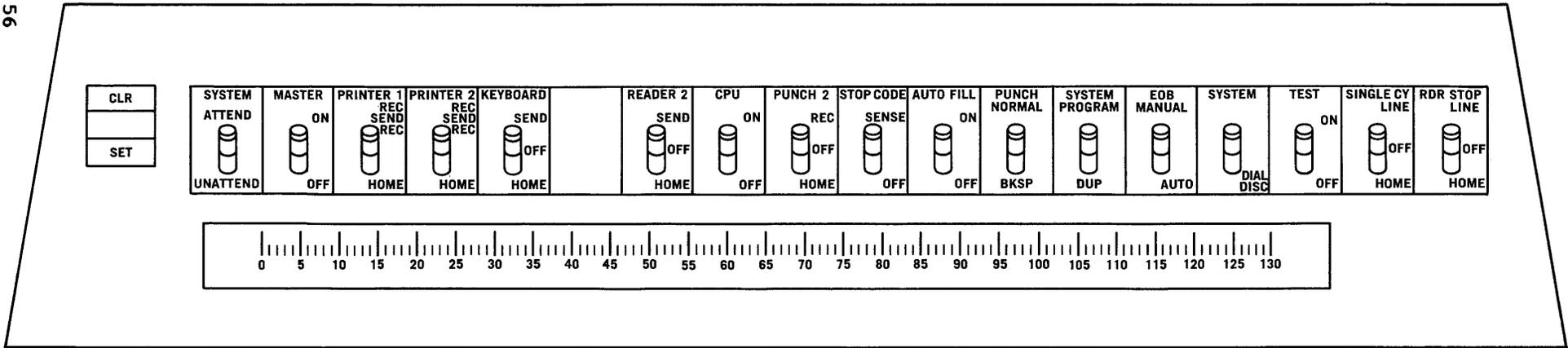


Figure 6. Model 3 Keyboard and Switch Panel

## HOW TO ENTER A COMMAND THROUGH THE IBM 1052 PRINTER-KEYBOARD

- Press the REQUEST key on the device.
- Wait for the PROCEED light to go on.
- Type the command.
- Signal EOB by holding down the alternate coding key and pressing the numeric 5 key.

Note: If the PROCEED light is on, do not press the REQUEST key. Type the command and signal EOB.

To correct errors, cancel the entire line by pressing the alternate coding key and the numeric 0 key; then retype the command. If an I/O error occurs, the console alarm will ring and you will have to start over. If your device can be backspaced, simple errors can be corrected by backspacing and retyping from the point of the error. See Figures 6, 7, 8, 9, and 10.

## HOW TO ENTER A COMMAND THROUGH A COMPOSITE CONSOLE

- Press the STOP key on the reader.
- Place the command card in the reader; no double slash, //, is used on a command card, as opposed to a command statement entered in the input stream.
- Press the START key on the reader.

## HOW TO SHIFT TO AN ALTERNATE CONSOLE DEVICE

- Issue the VARY command on the primary console to make the alternate console offline.
- Press the INTERRUPT key on the control panel.
- If a job was stopped by the INTERRUPT key, issue a CANCEL command for it.

## HOW TO ENTER A COMMAND INTO THE INPUT STREAM

Place the command statement card (or group of command statement cards) into a card reader, making sure it is followed by a JOB, EXEC, or null statement card.

Commands included in an MVT job stream may not take effect immediately. For

example, a STOP command included in the job stream to be stopped may not prevent a few of the cards that follow it from being read. The command will prevent any subsequent jobs from being placed in the job queue.

## HOW TO SPECIFY SYSTEM PARAMETERS (ALL SYSTEMS)

If you receive a SPECIFY SYSTEM PARAMETERS message when the nucleus is initialized, you may be required by your installation to alter several optional features of the system, such as the BLDL option, the RAM option, the RQ option, the RSVC option, the SQS option, the MIN option, the QBF option, or some combination of these options. If no changes are to be made, issue REPLY id, 'U'.

BLDL Option: This option specifies that the system is to have in main storage a directory of the link library -- a partitioned data set used in supplying load modules referred to in job control language statements or in macro instructions. To alter the option, issue REPLY id, 'BLDL=xx' where xx is a two-character change code supplied by your system programmer.

RAM Option: This option specifies that the system is to have certain frequently-used program modules placed into main storage. To alter the option, issue REPLY id, 'RAM=xx[,xx,xx,xx]' where each xx is a two-character code supplied by your system programmer.

RQ Option: This option specifies that the system is to have its input queue made resident in main storage. The option is not supported in systems with MVT. To alter this option, issue REPLY id, 'RQ=n' where n is a number telling how many input queue records should be resident. The number is supplied by your system programmer.

RSVC Option: This option specifies that the system is to make resident in main storage certain program modules, called non-resident SVC (supervisor call) routines, that normally would be kept on direct access storage. To alter this option, issue REPLY id, 'RSVC=xx[,xx,xx,xx]' where each xx is a two-character change code supplied by your system programmer.

SQS Option: This option specifies that a system with MVT is to have a system queue space of a certain size. To add to the size specified at the time the system was generated, issue REPLY id, 'SQS=n' where n is the number of 2048-byte blocks of storage to be added. The number is supplied by your system programmer.

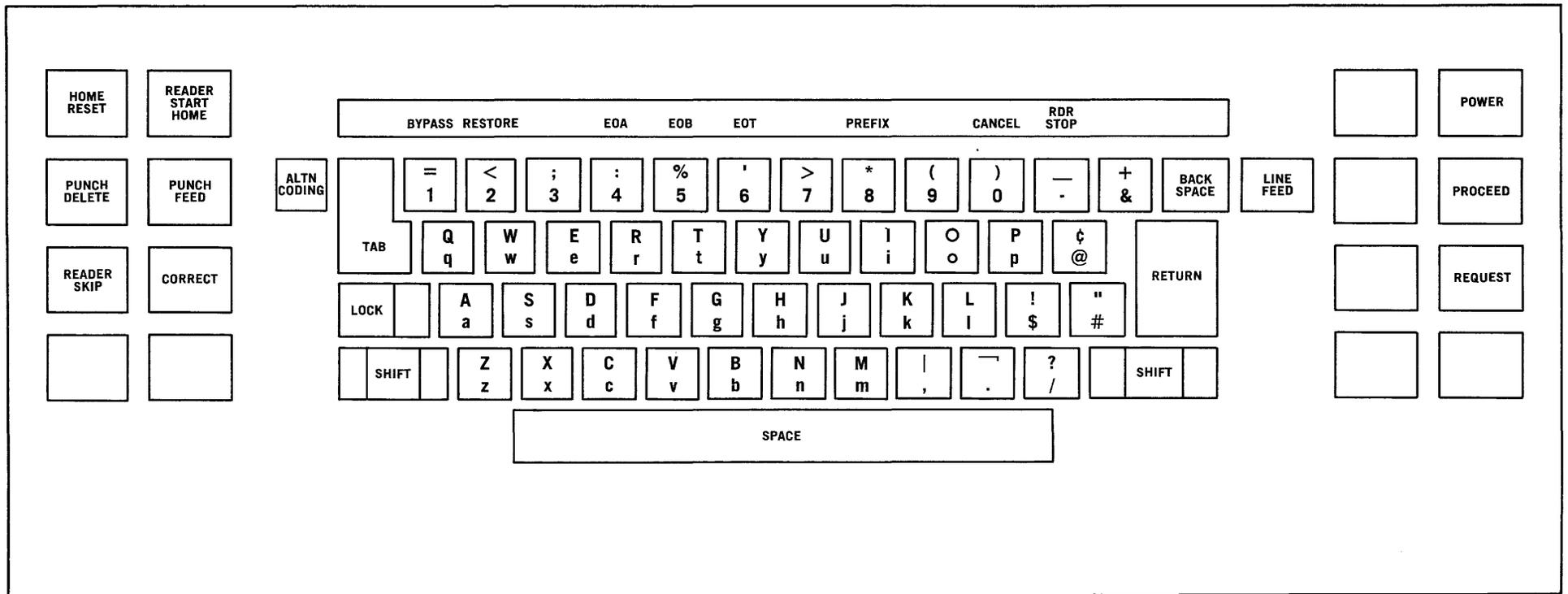
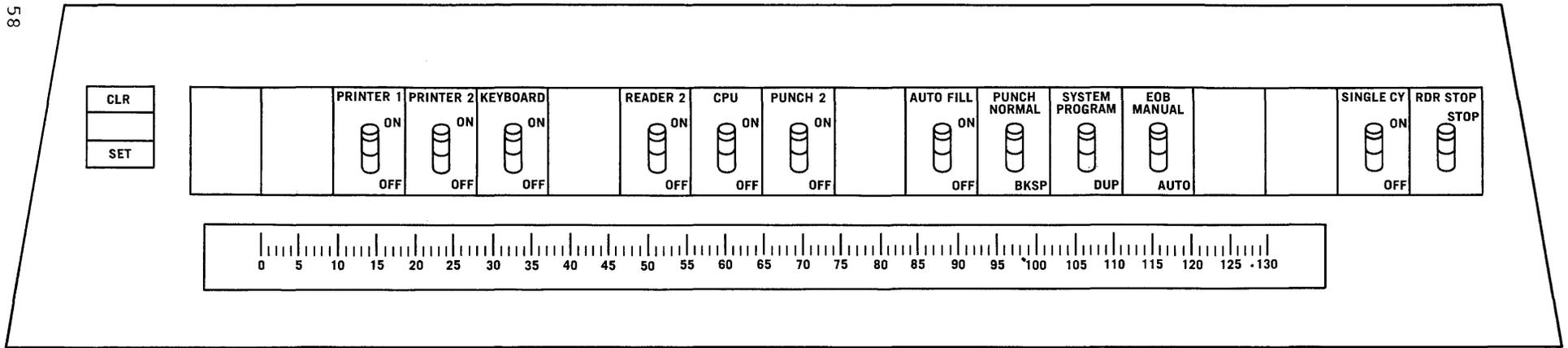


Figure 7. Model 5 Keyboard and Switch Panel

**MIN Option:** This option specifies that a system with MVT is to have a certain minimum-sized area of main storage to initiate each job. To override the size specified at the time the system was generated, issue REPLY id, 'MIN=n' where n is the minimum number of 1024-byte blocks to be used. The number is supplied by your system programmer, and will normally be 52 or greater.

**QBF Option:** This option specifies that a system with MVT is to have a buffer area of a certain size set aside for the input (job) queue. To override the size specified at the time the system was generated, issue REPLY id, 'QBF=n' where n is the number of 1024-byte blocks to be used. The number is supplied by your system programmer, and can be any number up to and including 255.

To cancel an option, reply with the option name, an equals sign, and a comma. For example, REPLY id, 'BLDL=,' cancels the BLDL option specified when the system was generated.

To get a list of the BLDL, RAM, and RSVC modules, issue REPLY id, 'U,L' -- or, if you wish to change an option and get a selective list or just get a list, reply with the option name, an equals sign, the change (or a 00), and an L. For example, REPLY id, 'RAM=00,L' will make the system print a list of the RAM modules.

To alter a combination of options, reply with the option names, equal signs, change parameters, and blanks. For example,

REPLY id, 'RAM=, BLDL=xx,L RSVC=00,L'

will cancel the RAM option, alter the BLDL option and give you a list of the BLDL modules, and list the unchanged RSVC modules. Please note that a blank must separate the keyword parameters as shown above.

If your reply to SPECIFY SYSTEM PARAMETERS is going to extend beyond one 80-character line, end the first line with CONT within the 80 characters. For example:

REPLY id, 'RAM=xx,xx,xx,xx,L CONT'

The system will then tell you to continue and then you can key in the rest of your reply:

REPLY id, 'BLDL=xx,L RSVC=xx,xx,xx,xx,L'

#### HOW TO LOAD A SECONDARY NUCLEUS

Ordinarily when you start the system, you are loading what is called the primary nucleus. If you are instructed to load a secondary nucleus, follow the same steps for starting a system with the primary control program but substitute the following for the second step.

- Set the ADDRESS COMPARE switch to stop the CPU on an equal compare at location 80 hex.
- Press the LOAD key.
- Set the ADDRESS ENTRY switches to location 8 hex.
- Enter a hex number, ranging from F1 to F9, into the left-most byte of the DATA switches so that location 8 will contain the actual character that will be included in the nucleus name.
- Set the STORAGE SELECT switch to store into MS.
- Press the STORE key.
- Turn off the ADDRESS COMPARE switch.
- Press the START key.

#### HOW TO LIMIT THE APPARENT SIZE OF MAIN STORAGE

Before loading the system, you can enter a value representing the highest main storage address in the system. This value can be less than the actual main storage limit. This causes any main storage locations higher than the limit to be inaccessible to the system.

The hex numbers you can enter, and the characters and maximum main storage sizes they represent are:

Hex Number	Character	Storage Size
C6	F	64K
C7	G	128K
C8	H	256K
C9	I	512K
D1	J	1024K

If you are instructed to limit the apparent size of main storage, proceed as follows:

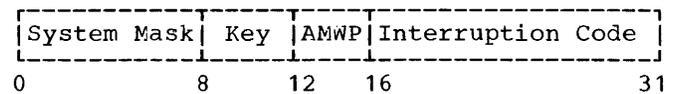
- Set LOAD UNIT to the direct-access storage device on which the operating system resides.
- Set ADDRESS COMPARE to stop the CPU at location 80 hex.

- Press LOAD.
- Set ADDRESS ENTRY to location 9 hex.
- Enter a hex number ranging from C6 to C9, or D1 into the left-most byte of the DATA switches so that location 9 will contain the actual character corresponding to the apparent maximum storage size.
- Press STORE.
- Turn off ADDRESS COMPARE.
- Press START.

where xxxxxx is the error code indicating the type of error condition. These codes, their explanations, and your responses are given in the publication IBM System/360 Operating System: Messages, Completion Codes, and Storage Dumps.

Displaying the PSW varies from one machine model to the next, so the steps are described according to model number.

Left Half of Current PSW

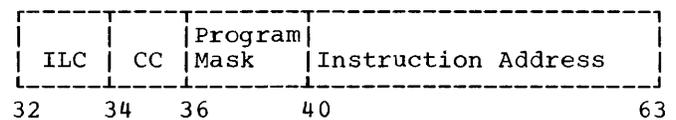


HOW TO DISPLAY THE CURRENT PSW

When the machine goes into a WAIT state, certain codes can be displayed in the second half of the current PSW to give an indication of what happened. The second half of the PSW looks like this:

00xxxxxx

Normal Right Half of Current PSW



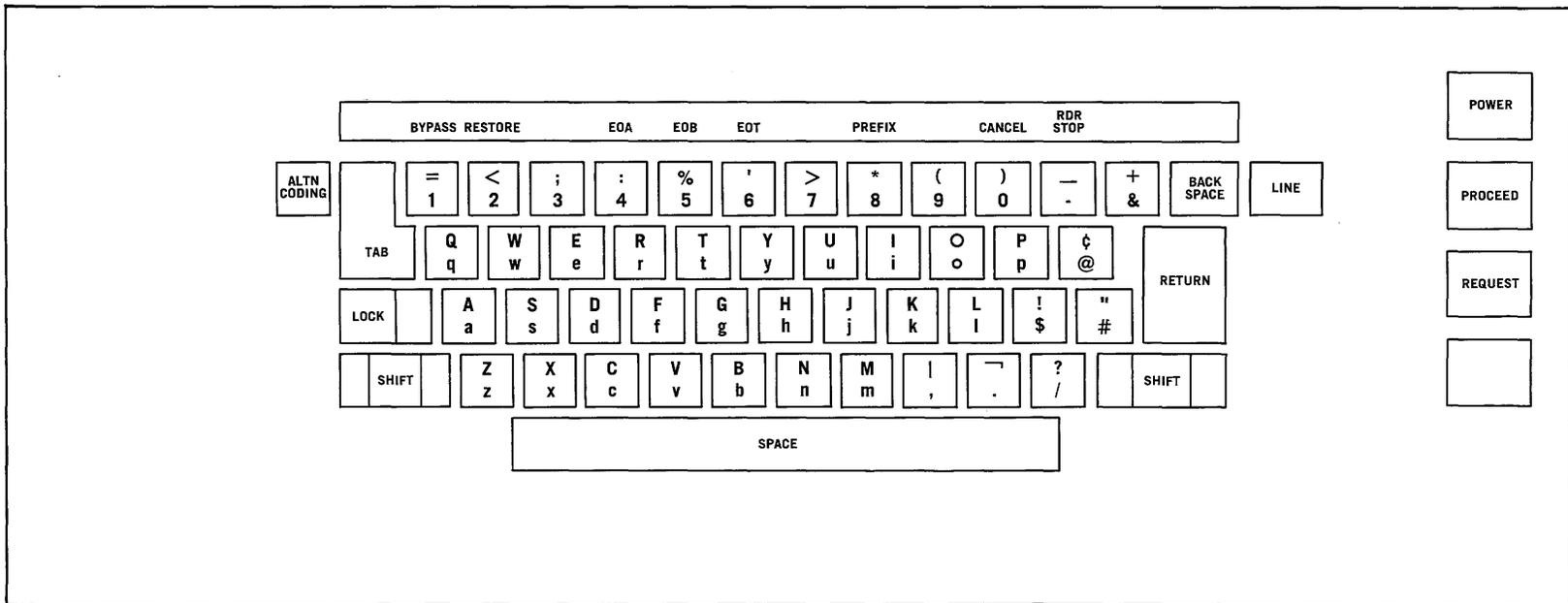
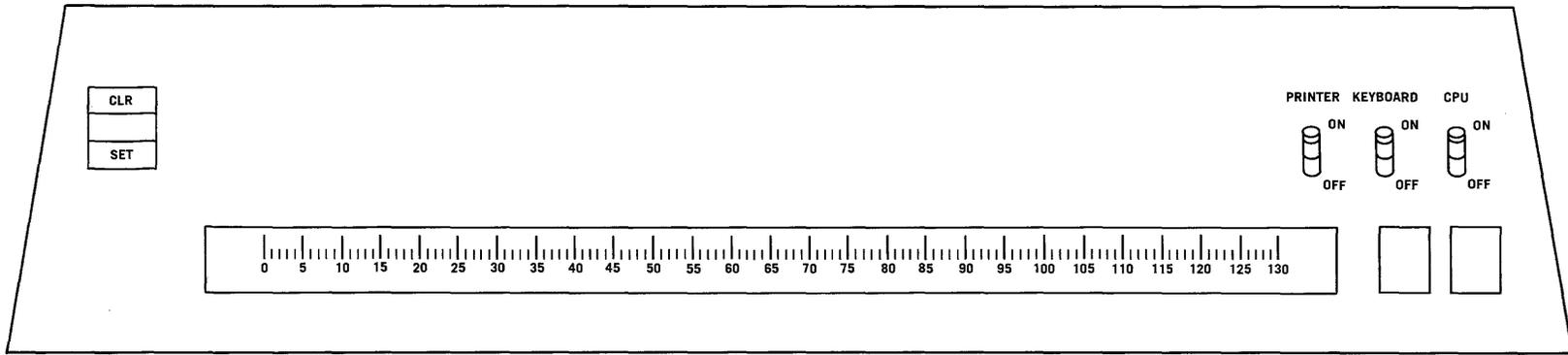


Figure 8. Model 6 Keyboard and Switch Panel

### On the Model 30

Storage displaying, including displaying the current PSW, is done one byte at a time on the Model 30 (Figure 11).

- Press STOP.
- Turn the STORAGE SELECT switch E to 'AS' with red flag up.
- Turn switch A to 'LS' or numeric 7.
- Turn MAIN STORAGE switches: B to zero, C and D to the desired location. PSW fields will be found at the locations below.

System Mask	B8
Protect key	B9 (high order 4 bits)
AMWP	B9 (low order 4 bits)
Interruption Code	External old PSW (bits 16-31)
Instruction Length	AC (bits 4 and 5)
Condition Code	BB (high order 4 bits)
Program Mask	BB (low order 4 bits)
Instruction Address	A9 & AA (or I & J register)

Note: Condition code is displayed as four bits (8, 4, 2, 1), one bit at a time.

- Press DISPLAY to display the data in the main storage data register and the address in the low order eight bits of the main storage address register.

### On the Model 40

Storage displaying, including displaying the current PSW, is done two bytes at a time on the Model 40 (Figure 12).

- Press STOP.
- Set the STORAGE SELECT switch to PSW.
- Set the STORAGE ADDRESS bit switches as follows:

All bits off for the first half-word of the PSW.  
Bit 7 on for the second half-word.  
Bit 6 on for the third.  
Bits 6 and 7 on for the last half-word.

- Press the DISPLAY key.

### On the Model 50

Storage displaying, including displaying the current PSW, is done 4 bytes (one word) at a time on the Model 50 (Figure 13).

- Press STOP.
- Set the ADDRESS switches to 170.
- Set the STORAGE SELECT switch to LOCAL.
- Press the DISPLAY key.
- Rotate roller 3 (CPU 1) to position 1, showing the L register which contains the first half of the PSW.
- Rotate roller 4 to position 3 and examine bits 6-13, labeled PSW, to find the first byte of the second word of the PSW.
- The last three bytes of the PSW are in the instruction address register.

### On the Model 65

Storage displaying, including displaying the current PSW, is done eight bytes (two separate words) at a time on the Model 65 (Figure 14).

- Press STOP.
- Look at roller 4 position 1 for the left half of the current PSW.
- Look at the D register (roller 1 position 2) for the right half.

### On the Model 75

Storage displaying, including displaying the current PSW, is done eight bytes (one whole double-word) at a time on the Model 75 (Figure 15).

- Press STOP.
- Look at the PSW register indicator lights for the current PSW.

### HOW TO BYPASS A CONSOLE MALFUNCTION

If a console malfunction occurs, the system rings the console alarm and retries the operation. If the malfunction is not corrected after a number of retries, systems with the primary control program enter the wait state with an error code of either 000E00 or 000E01 in the right half of the current PSW. If you do not want to repeat the IPL procedure, you can attempt a recovery:

- If you want to shift to another console before resuming operation, press the INTERRUPT key on the system control panel.
- Press STOP on the console.

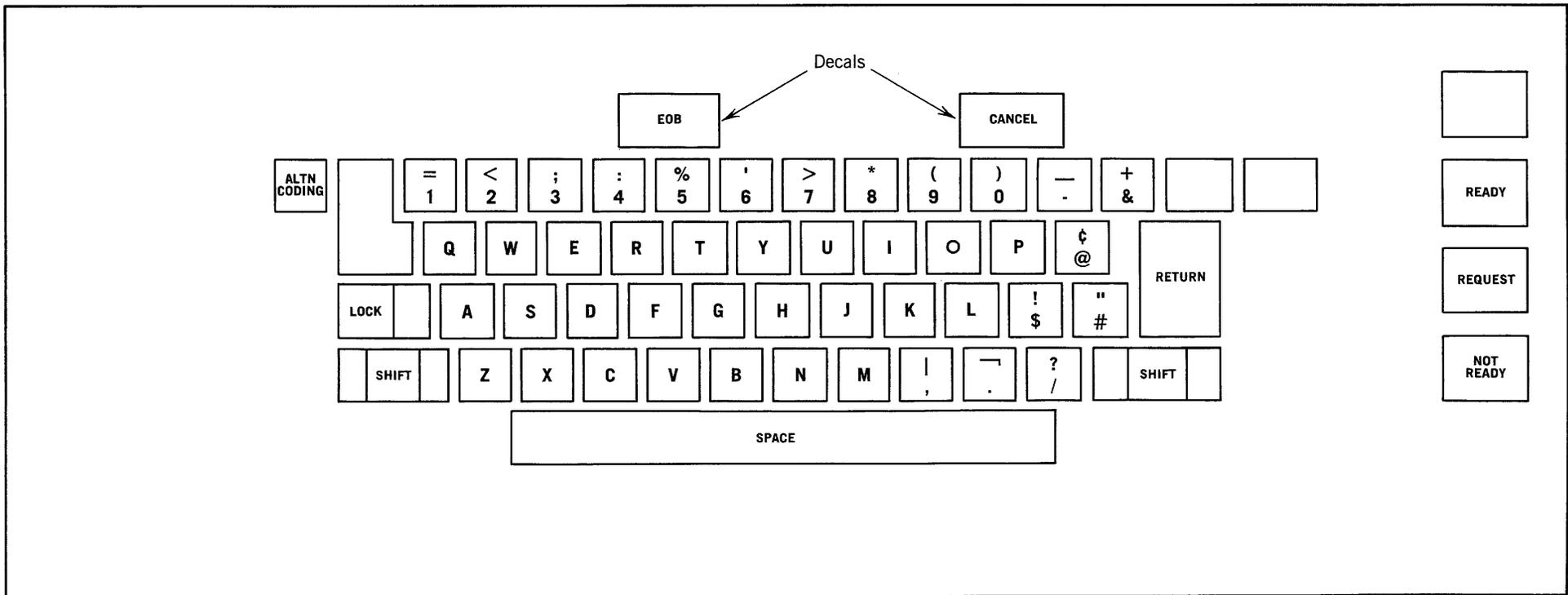
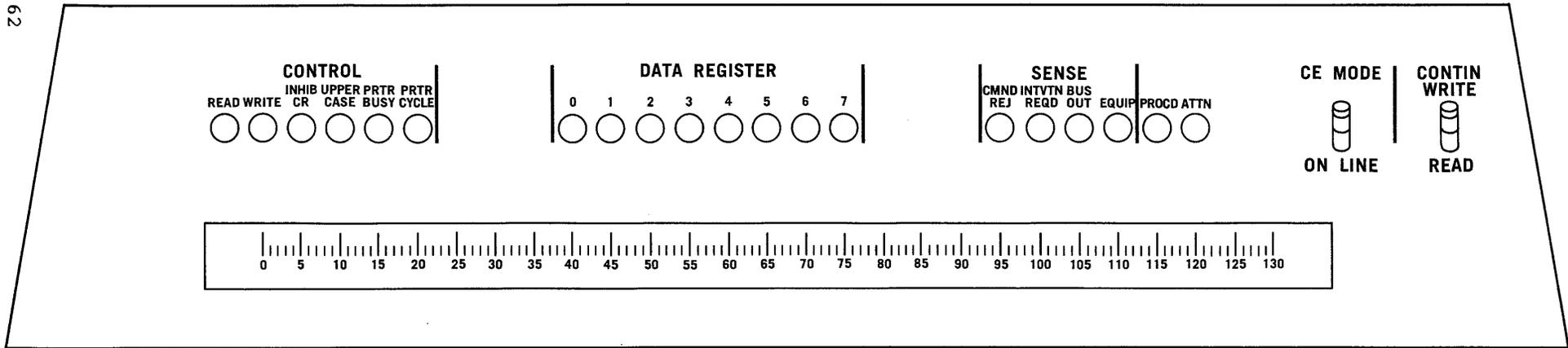


Figure 9. Model 7 Keyboard and Switch Panel

- Put a zero in the wait bit -- bit 14 -- and zeros in the instruction address field -- bits 40-63 -- of the current PSW.
- Press START.

The system resumes operation and retries the operation involved in the malfunction. If you pressed the INTERRUPT key, the retry takes place on the new console.

In systems with MFT, and in systems with MVT, 000E02 is loaded for all 1052 errors, while 000E01 is loaded for all printer errors. If you want the error to be ignored put 4 in register 13. If you don't alter register 13, the system will retry the error-causing operation. (Register 14 contains a resumption address, subject to modification by the contents of register 13.)

Model 65 Note: If you do not wish to repeat the IPL procedure on a Model 65, you can attempt a recovery with the following procedure:

- Press the console STOP key.
- Put the left-most four bytes of the current PSW into location 0 and zero out the wait bit - bit 14.
- Put the four bytes of register 14 into location 4.
- Press PSW RESTART.

#### HOW TO DISPLAY MAIN STORAGE

Very often, you will be called upon to display and record various locations in main storage, particularly such permanent storage areas as the old and new PSWs, the channel address word (CAW), and the channel status word (CSW). The addresses for these controlling words, and their formats, are given after the description of the storage-displaying procedure on the Model 75.

#### On the Model 30

- Press STOP.
- Turn the STORAGE SELECT switch E to 'MS' with the red flag up.
- Turn the main storage address switches A, B, C, and D to the address to be displayed.
- Press DISPLAY. The data will be in the main storage data register, while the address will be in the main storage address register.

#### On the Model 40

- Press STOP.
- Set the STORAGE SELECT switch to 'MS'.
- Set the STORAGE ADDRESS switches to the address to be displayed.
- Press DISPLAY. The data will be in the STORAGE DATA registers, bytes 0 and 1.

#### On the Model 50

- Press STOP.
- Set the address switches to the address to be displayed.
- Set the STORAGE SELECT switch to MAIN.
- Press DISPLAY. The data will be in the storage data register.

#### On the Model 65

- Press STOP.
- Set the STORAGE SELECT lever switch to the middle position -- main storage.
- Set the address switches to the address to be displayed.
- Press DISPLAY. The data will be in the ST register (roller 1 position 3 and roller 2 position 3) and the AB register (roller 3 position 3 and roller 4 position 3).

#### On the Model 75

- Press STOP.
- Set the STORAGE SELECT switch to MAIN STOR.
- Set the address switches to the address to be displayed.
- Press DISPLAY. The data will be in the J register.

#### Permanent Main Storage Assignments

Hex Address	Length	Content
0	8 bytes	Initial program loading PSW
8	8 bytes	Initial program loading CCW1
10	8 bytes	Initial program loading CCW2
18	8 bytes	External old PSW

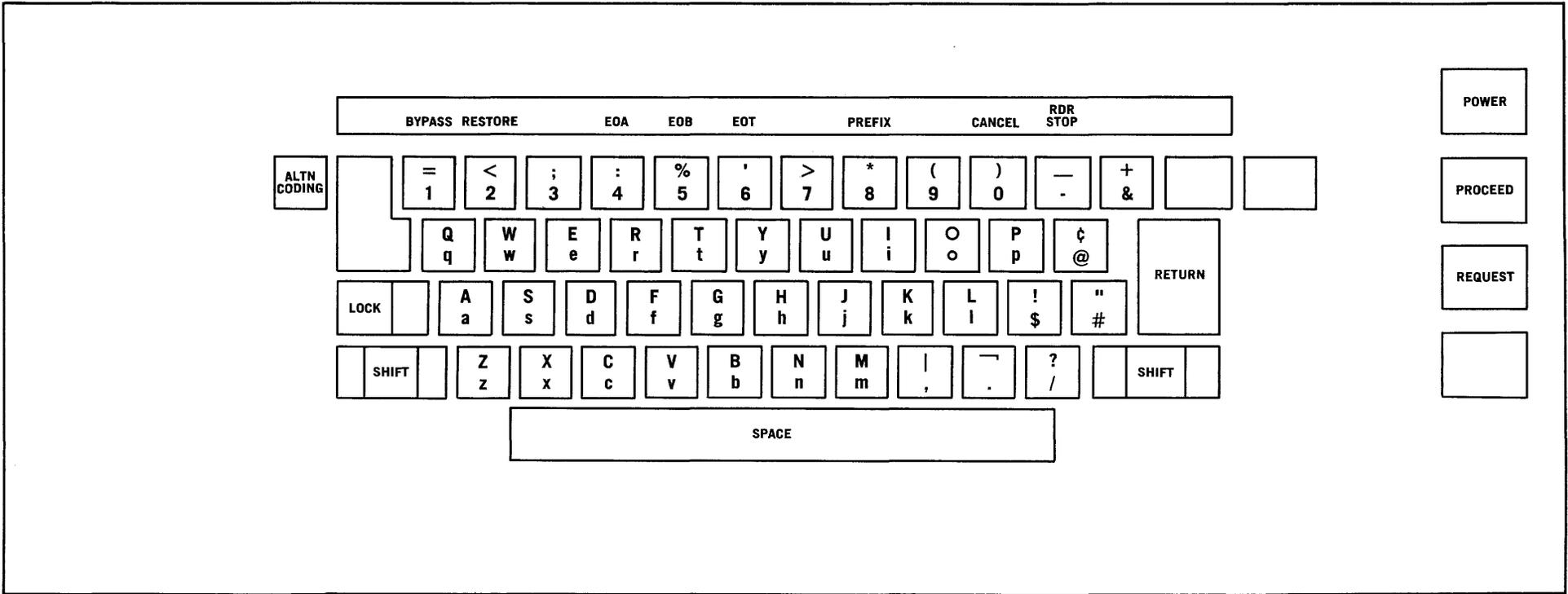
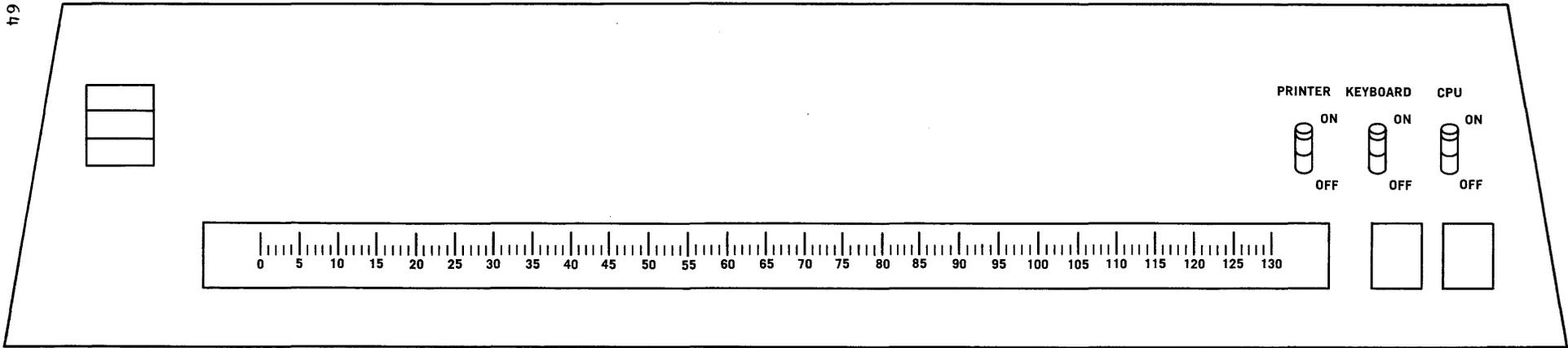


Figure 10. Model 8 Keyboard and SWITCH Panel

Hex Address	Length	Content
20	8 bytes	Supervisor call old PSW
28	8 bytes	Program old PSW
30	8 bytes	Machine check old PSW
38	8 bytes	Input/output old PSW
40	8 bytes	Channel status word
48	4 bytes	Channel address word
4C	4 bytes	Unused
50	4 bytes	Timer
54	4 bytes	Unused
58	8 bytes	External new PSW
60	8 bytes	Supervisor call new PSW
68	8 bytes	Program new PSW
70	8 bytes	Machine check new PSW
78	8 bytes	Input/output new PSW
80	up to 256 bytes	Diagnostic scan-out area

Program Status Word

The PSW contains information required for program execution. By storing the PSW, the control program can preserve the status of the CPU for later inspection. By loading a new PSW or part of a PSW, the status of the CPU can be changed.

The format of old and new PSWs is the same as that of the current PSW, shown under the earlier heading "How to Display the Current PSW."

The interruption code (bits 16-31) in the old PSWs indicates the source of the most recent interruption.

External Old PSW -- Hex location 18

Code	Interruption Source
0001	External Signal 7
0002	External Signal 6
0004	External Signal 5
0008	External Signal 4
0010	External Signal 3
0020	External Signal 2
0040	Interrupt Key
0080	Timer

External interruptions from more than one source can occur at one time. An interruption code of 008A means, for example, that an interruption was requested by the timer and external sources 4 and 6.

Supervisor Call Old PSW -- Hex location 20

Code	Interruption Source
00xx	SVC instruction in a program

In the interruption code, xx is the I field of the SVC instruction that was given.

Program Old PSW -- Hex location 28

Code	Interruption Source
0001	Op code incorrect
0002	Privileged operation
0003	Execute error
0004	Protection
0005	Addressing
0006	Specification
0007	Data
0008	Fixed point overflow
0009	Fixed point divide
000A	Decimal overflow
000B	Decimal divide
000C	Exponent overflow
000D	Exponent underflow
000E	Significance
000F	Floating point divide

Machine Check Old PSW -- Hex location 30

Code	Interruption Source
0000	CPU or channel error

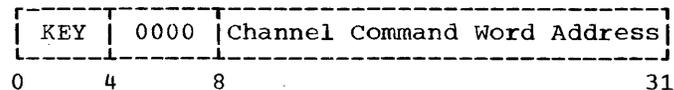
Input/Output Old PSW -- Hex location 38

Code	Interruption Source
00xx	Multiplex channel
01xx	Selector channel 1
02xx	Selector channel 2
03xx	Selector channel 3
04xx	Selector channel 4
05xx	Selector channel 5
06xx	Selector channel 6

In the interruption code, xx is the control unit and device address.

Channel Address Word

The CAW specifies the storage protection key and the address of the first channel command word associated with the START I/O instruction. The CAW is found at hex location 48.



Channel Command Word

The CCW specifies the command, the storage area to be used for I/O operations, and the action to be followed when the operation is completed. CCWs can be anywhere in main storage, and can exist singly or in a group called a channel program.

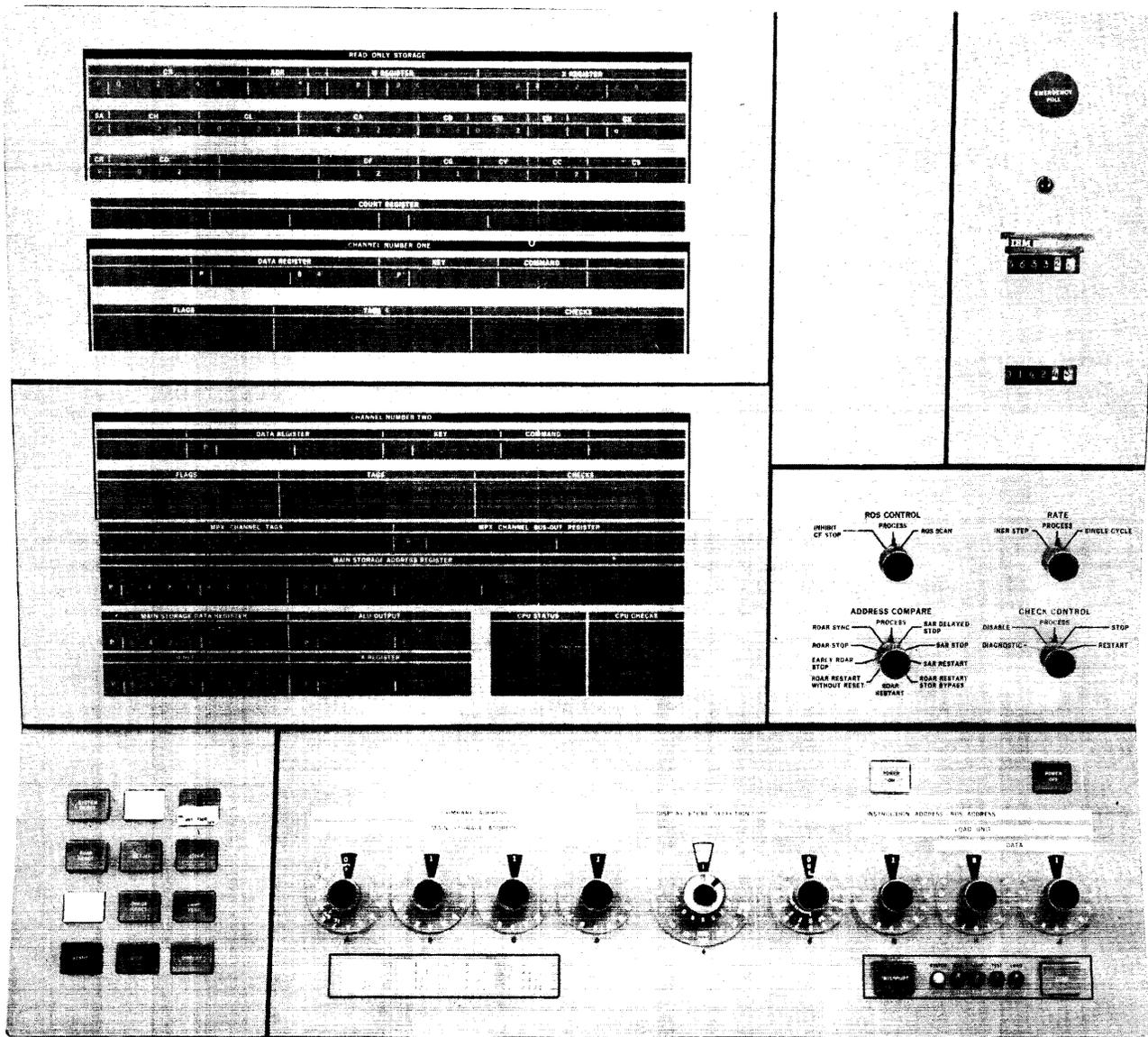
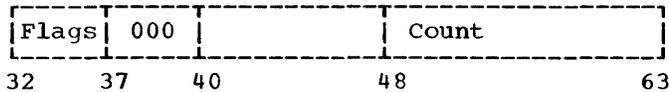
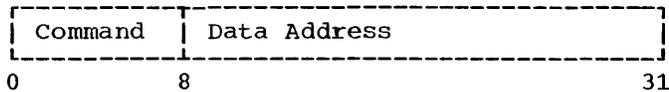


Figure 11. Model 30 System Control Panel



Bits 0-7 give the command code (m identifies a modifier bit, while x indicates that the bit position is ignored):

mmmm0100	Sense command
xxxx1000	Transfer in channel command
mmmm1100	Read backward command
mmmmmm01	Write command
mmmmmm10	Read command
mmmmmm11	Control command

Bits 8-31 give the location of a byte in main storage.

Bit 32 causes the address portion of the next CCW to be used.

Bit 33 causes the command code and data address in the next CCW to be used.

Bit 34 causes a possible incorrect length indication to be suppressed.

Bit 35 suppresses the transfer of data into main storage.

Bit 36 causes a program-controlled interruption (PCI).

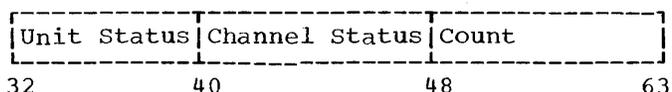
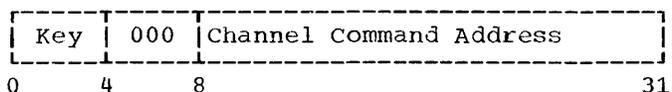
Bits 37-39 contain zeros.

Bits 40-47 are ignored.

Bits 48-63 specify the number of bytes in the operation.

#### Channel Status Word

The CSW provides information about the termination of an input/output operation. The CSW is found at hex location 40. It can be made up or changed by input/output interruptions as well as by START I/O, TEST I/O, and HALT I/O instructions.



Bits 0-3 contain the protection key used in the last operation.

Bits 4-7 contain zeros.

Bits 8-31 contain the address plus 8 of the last CCW used.

Bits 32-39 contain the unit status byte:

Bit 32	-- attention
Bit 33	-- status modifier
Bit 34	-- control unit end
Bit 35	-- busy
Bit 36	-- channel end
Bit 37	-- device end
Bit 38	-- unit check
Bit 39	-- unit exception

Bits 40-47 contain the channel status byte:

Bit 40	-- program-controlled interruption
Bit 41	-- incorrect length
Bit 42	-- program check
Bit 43	-- protection check
Bit 44	-- channel data check
Bit 45	-- channel control check
Bit 46	-- interface control check
Bit 47	-- chaining check

Bits 48-63 contain the residual count of the last CCW used.

#### HOW TO ANALYZE INPUT/OUTPUT COMMANDS

- Look at the channel address word (CAW) at location 48 hex to find the channel command word (CCW). If command chaining was used, the CAW points to the first CCW in the chain.
- Look at the channel status word (CSW) at location 40 hex:
  1. The command address (bits 8-31) points to the last CCW executed plus 8 bytes.
  2. The status portion (bits 32-47) tells the status of the channel control unit or subchannel, and the status of the device the command was issued to (each device has its own meanings for the status bits -- see your hardware manuals). The address of the device is found in bits 16-31 of the I/O old PSW at location 3A hex.



3. The residual byte count should be zero. If it is not, one of three things is indicated: a wrong-length record was met, indicating a problem with the channel program; the command was rejected by the channel, also indicating a fault in the channel program; a data check during a read or write stopped data transfer and device motion. Channel end, device end, unit check, and incorrect length indications are in the CSW, and the residual byte count may show how much data was not transferred.

4. When working with variable-length records, the wrong-length indicator (bit 34) should be on to prevent I/O interruptions.

- Look at the channel command word (CCW) to find the data address, byte count of the data, command code naming the actual operation, and flag bits for command and data chaining. Except for the transfer-in-channel (TIC) command, there must be a byte count of one or more for any I/O operations.

#### HOW TO USE INDEPENDENT UTILITY PROGRAMS

Independent utility programs operate outside the operating system. They include:

- DASDI (IBCDASDI), a program that initializes and assigns alternate tracks to a direct-access volume.
- DUMP/RESTORE (IBCDMPRS), a program that dumps and restores the contents of a direct-access volume.
- RECOVER/REPLACE (IBRCVPR), a program that recovers usable data from a bad track, assigns an alternate track, and merges replacement data with the recovered data onto the alternate track.

Independent utilities are loaded as card decks or as card images on tape. The programs and the control statements needed to request them are described in the publication IBM System/360 Operating System: Utilities. To run the programs:

- Put the deck in the reader or mount the tape reel that contains the program.
- Load the program from the reader or tape drive by setting the load selector switches and pressing the console LOAD key. When the program is loaded, the system goes into a wait state and the console lights display hex FFFF.

- If you have a console typewriter, press its REQUEST key. When DEFINE INPUT DEVICE is printed, type in

INPUT=xxxx cuu

where xxxx is the device type, c is the channel address, and uu is the unit address. (The device type can be 1402, 1442, 2400 or 2540. In addition, the RECOVER/REPLACE program can use the 1052 as a device type.)

- If you do not have a console typewriter, enter one of the following numbers at hex location 0110.

1cuu for a 1442 card read punch.  
 2cuu for a 2400 nine-track magnetic tape drive.  
 0cuu for a 1402 or 2540 card read punch.

Press the console INTERRUPT key and the START key.

- When the job is done, the program prints out END OF JOB and enters the wait state. (If the program was RECOVER/REPLACE and the message output device is a tape, the console lights display the hex value DDDD for a normal end of job, and EEEE for an abnormal end.)

#### HARDWARE DEBUGGING AIDS

System environment recording (SER) is a set of control program routines that record hardware bugs in the CPU and channels in Models 40, 50, 65, and 75. SER includes two optional, model-dependent programs called SER0 and SER1. Your installation may have either SER0 or SER1, or neither.

Always run the system with the check control switch (Models 30 and 50) or the CPU check switch (Models 40, 65, and 75) set to PROCESS. This setting is required by SER routines, as well as by a standard, stand-alone program called SEREP (system environment recording edit and print).

When a machine check interruption occurs while the CPU check switch is in process mode, either the system puts itself into a wait state or gives control to SER. If the system goes into a wait, run SEREP as described under the next heading.

If control goes to SER0, it figures out the type of bug and, if possible, writes out a record describing the error on a data set called SYS1.LOGREC. This data set resides on the primary residence volume.

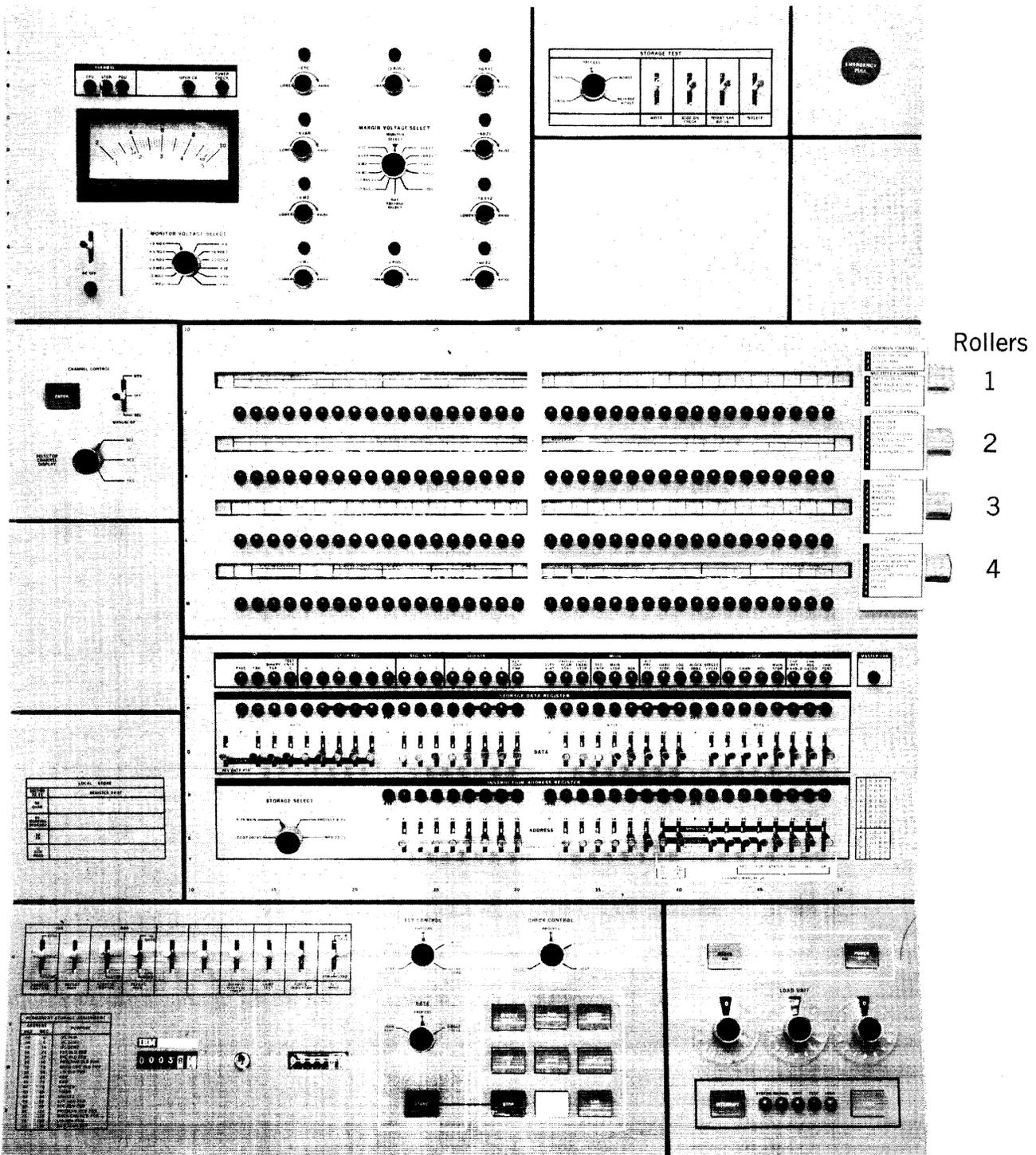


Figure 13. Model 50 System Control Panel

If SER0 cannot write the record, the system goes into a wait state and a message is printed out on the keyboard telling you to use SEREP. If the recording is partially or fully done, the system goes into a wait state and a message is printed out telling you to reload the operating system.

If control goes to SER1, it also collects and writes out hardware data, but in addition, it attempts to link the error with a particular piece of work being done. If the error can be linked with a particular piece of work and if the control program has not been damaged by the error, that piece of work is terminated abnormally; otherwise, the system goes into a wait state.

When the SYS1.LOGREC data set has been filled -- by either of the SER versions plus other debugging programs such as OBR (outboard recorder) and SDR (statistical data recorder) -- print out the data set's contents by using the environment recording edit and print routine, called IFCEREPO. This routine formats and writes the records on SYS1.LOGREC onto printer, tape, or disk according to your installation's specifications. IFCEREPO is described in the publication IBM System/360 Operating System: Utilities.

To use IFCEREPO to write out the contents of the SYS1.LOGREC data set and at the same time clear the data set for re-use, you can use a JOB card followed by:

```
//          EXEC    PGM=IFCEREPO
//SERLOG   DD      DSNAME=SYS1.LOGREC,      x
//          DISP=(OLD,KEEP)
//EREPPPT DD      SYSOUT=A
```

#### How to use SEREP

When a message or PSW code indicates that you should run the SEREP program to print out debugging information:

- Load the SEREP deck in the card reader.
- Set the LOAD UNIT switches to the address of the card reader.
- Press the LOAD button on the system control panel.
- Save the SEREP printout for later use by your customer engineer.
- Reload the operating system.

If you are repeatedly asked to run the SEREP program, call your customer engineer.

#### CREATING VOLUME LABELS

For a description of a program you can use to create volume labels on standard-label tapes, see the section on the program IEHINITT in the publication IBM System/360 Operating System: Utilities.

#### UNIVERSAL CHARACTER SET (UCS) UTILITY PROGRAM

The Universal Character Set (UCS) feature provides for printing any set of 240 graphics by the 1403 printers attached to the 2821 control unit. The 2821 has one 240 character read/write storage unit for each printer. Each storage position corresponds to a character on the train or chain of the respective printer. The UCS program stores a requested set of graphics into the 2821 control unit.

By pre-loading chain or train cartridges before an explicit console request is received from the program, and by scheduling together several programs requiring the loading of the universal character set, throughput can be increased.

The identifying names for UCS print chains and print trains are defined by each installation and should be familiar to the programmers and machine room schedulers as well as to operators, since the running of the UCS program depends on using the IDs correctly.

For a description of the UCS program, see the publication IBM System/360 Operating System: Utilities.

For more information about character sets and loading of print train and print chain cartridges, see the publications IBM 2821 Control Unit, Form A24-3312 and IBM 1403 Printer, Form A24-3073.

#### CONTROL STATEMENTS

Control statements provide you with another means of controlling the operating system. They are, in effect, orders to the system to perform certain functions.

Prescribed formats must be followed in using control statements; incorrect usage might result in failure of a job to enter the system, abnormal termination of a job, or incorrect output.

For detailed information on how to use control statements, refer to IBM System/360 Operating System: Job Control Language, and to IBM System/360 Operating System: Job Control Language Charts.

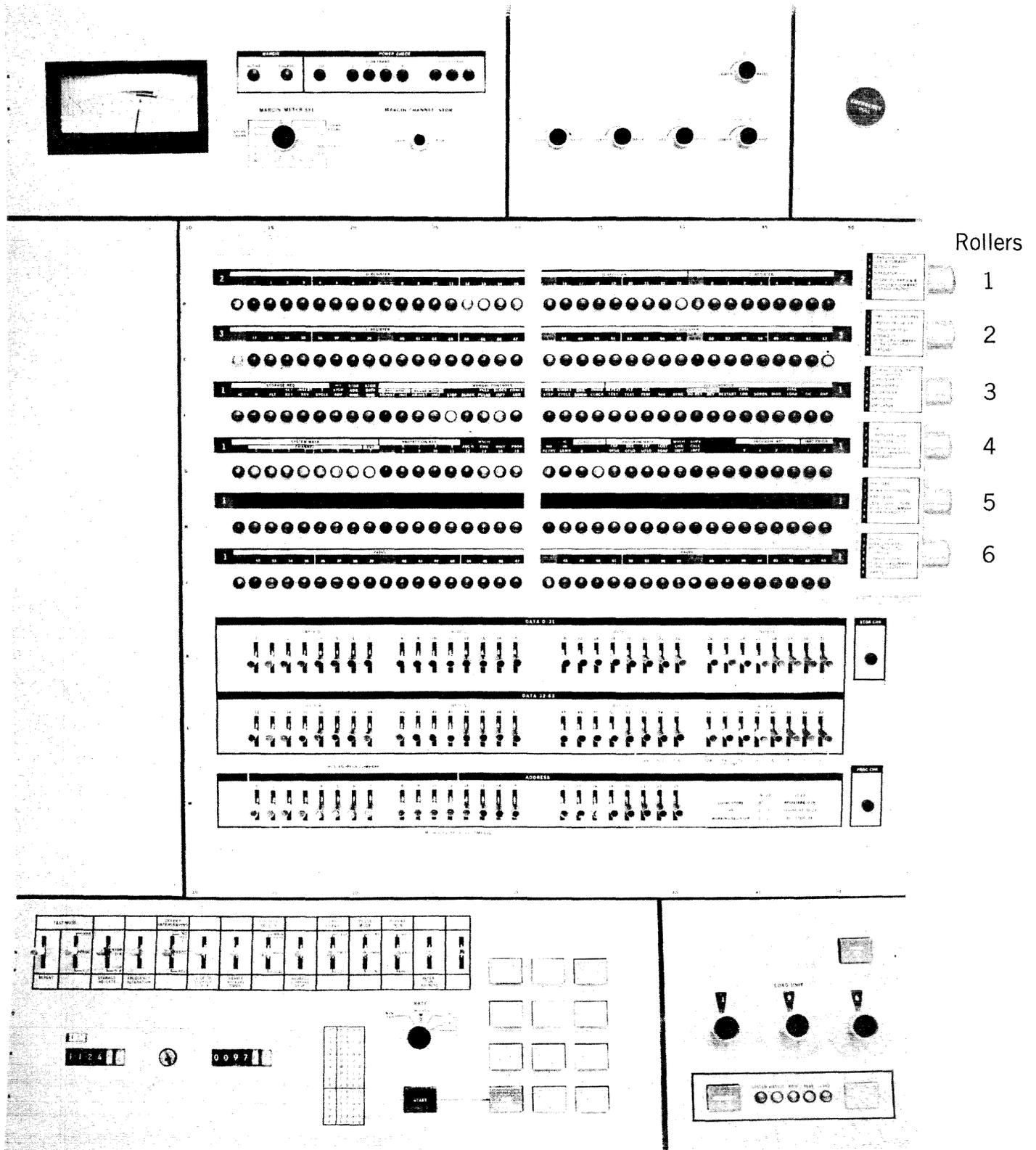


Figure 14. Model 65 System Control Panel

Of the various kinds of statements used with the operating system, you most frequently see the control statements that are interpreted and executed by the job scheduler. Most of these are originated by the programmer. Two of the statements, however, are useful to you: the null and the command statements.

The null statement can be used to mark the end of any job in the input stream, and the command statement is used to enter operator commands through an input device other than a console typewriter.

All control statements for the job scheduler are defined by the initial characters // (two slashes) or /\* (a slash followed by an asterisk). Like operator commands, control statements are free-form.

The six job scheduler control statements are described in this section. A dotted line within the format illustrations indicates that other information will follow in the statement.

#### Null Statement

The null statement is used to mark the end of the last job in an input stream. If it appears elsewhere in the input stream, it acts as a terminal statement of the job in which it occurs. There are no parameters or options for this statement. It begins with two slashes (//); the rest of the statement contains blanks. The null statement format is:

```
-----  
//  
-----
```

#### Command Statement

The command statement is used to enter commands by means of an input device other than the console device. The command must be preceded by two slashes (//) followed by at least one blank.

Like operator commands, command statements may not be continued; that is, the complete statement must be contained in one logical record. A command statement must precede another control statement. The command statement format is:

```
-----  
// command statement  
-----
```

#### Job Statement

The job statement is the first statement of a job. It indicates to the operating system the beginning of a new job and the end of the previous job. This statement usually contains such information as the programmer's account number, his name, the job priority, etc. Many of its fields are optional. The job statement format is:

```
-----  
//jobname JOB...  
-----
```

#### Execute Statement

The execute statement is the first statement of each job step within a job. It usually contains such information as the step name, program name, maximum running time permitted for the job step, parameters to be passed to the program, accounting information, etc. Many of its fields are optional. The execute statement format is:

```
-----  
//stepname EXEC...  
-----
```

#### Data Definition Statement

The data definition statement allows you to issue instructions to the operating system regarding the definition and disposition of the data sets used in each step. All data definition statements for a particular step immediately follow the execute statement for that step, with one exception: if the job executes programs in a private library, a data definition statement with the ddname JOBLIB must appear immediately before the first execute statement of the job. The data definition statement format is:

```
-----  
//ddname DD...  
-----
```

#### Delimiter Statement

The delimiter statement is used to separate SYSIN data from the control statements that follow it. It is not necessary to use this statement in systems with MVT. The delimiter statement format is:

```
-----  
/*  
-----
```

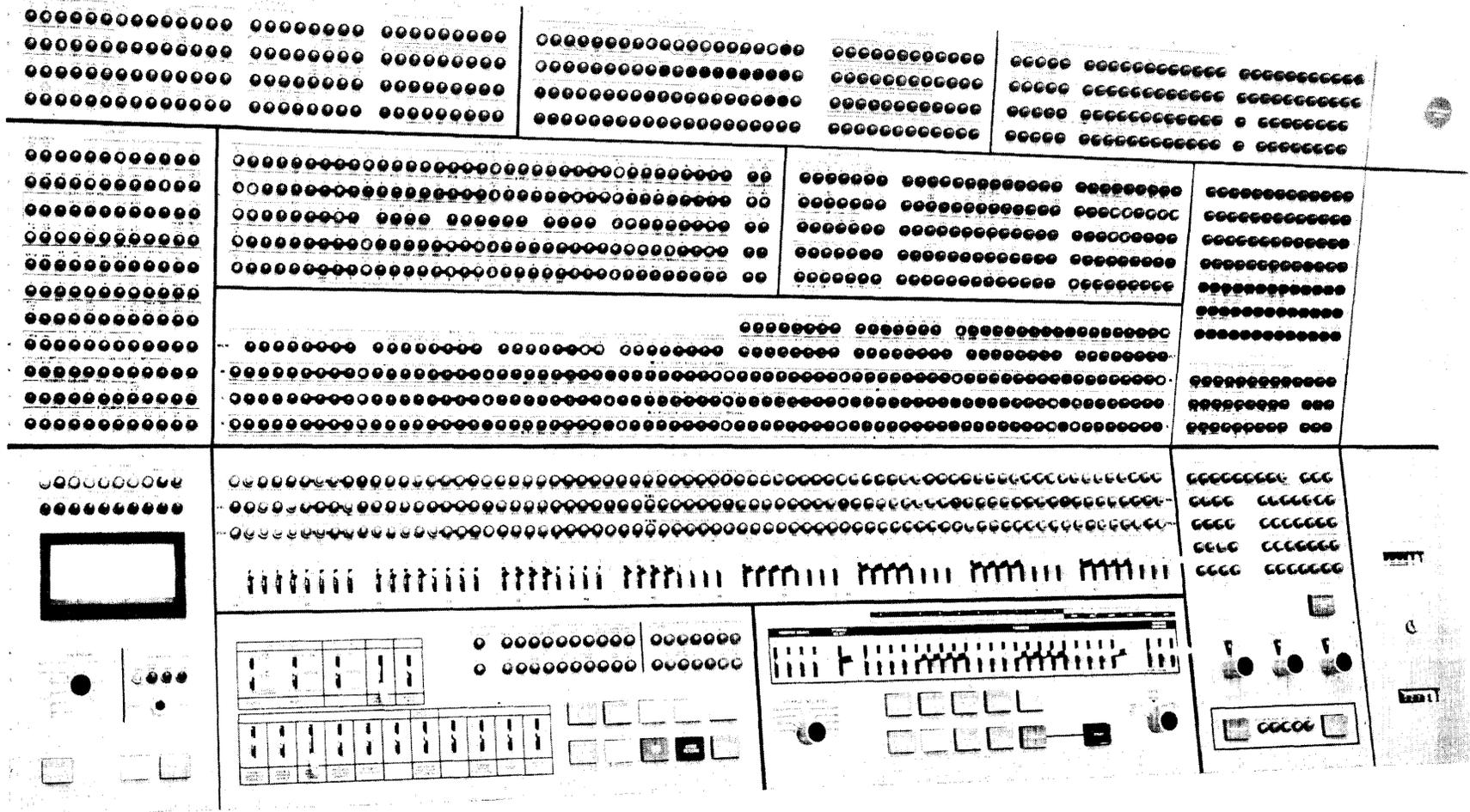


Figure 15. Model 75 System Control Panel

## ADDITIONAL OPERATING TECHNIQUES

Preceding sections have described how to display the PSW and main storage. This section describes additional techniques used in operating System/360 Models 30, 40, 50, 65, and 75.

### MODEL 30

#### Clearing Main Storage:

- Press STOP.
- Set instruction address keys to 0BF9.
- Set ROS CONTROL to ROS SCAN.
- Set CHECK CONTROL to DISABLE.
- Press SYSTEM RESET.
- Press ROAR RESET.
- Press START.
- To stop the clearing process, set RATE switch to SINGLE CYCLE.
- Press SYSTEM RESET.

#### Altering Main Storage:

- Set up address as you would in displaying main storage (machine must be in manual mode).
- Set the byte of data to be stored into the rotary data switches H and J (use hexadecimal representation of data).
- Press STORE; the new data is displayed in the main storage data register.

#### Displaying a General Purpose Register (GPR):

- Machine must be in manual mode.
- Set switch A to LS.
- Set switch B to zero.
- Set switch C to desired GPR.
- Set switch D to correspond to desired byte of GPR - that is, to 0, 1, 2, or

3. The data is displayed in the main storage data register.

#### Displaying a Floating Point Register (FPR):

Floating point registers are an optional feature. If your machine has this feature, you can display an individual FPR following the same procedure used in displaying a GPR, with the following differences.

- Set switch C to desired FPR.
- Set switch D to correspond to desired byte of FPR:

<u>FPR byte</u>	<u>Switch D setting</u>
0	8
1	9
2	A
3	B
4	C
5	D
6	E
7	F

- The data is displayed in the main storage data register.

#### Altering Registers:

- Follow same procedure as in displaying a register.
- Put the data you wish to enter into the storage data portion of the keys.
- Press STORE.

#### Stopping on Address Compare:

- Put the address you wish to stop on in address switches A, B, C, and D.
- Put ADDRESS COMPARE switch on SAR DELAYED STOP.
- The CPU status MATCH indicator will come on when the address is reached.

#### Altering The Current PSW:

- Proceed as in displaying the current PSW.
- Put new data in the data keys.
- Hit STORE and the new data will be entered.

Clearing Main Storage:

- Press STOP.
- Disable the interval timer.
- Press SYSTEM RESET.
- Set the RATE switch to SINGLE CYCLE.
- Enter 1000 (hexadecimal) in STORAGE DATA keys.
- Flip up the STORAGE STATS switch.
- Set RATE switch to PROCESS.
- Set DIAGNOSTIC CONTROL switch to MS ADDRESS.
- Press START. The microprogram light should come on when main storage is cleared. If any other red lights are on, main storage is not fully cleared; repeat procedure.
- Turn DIAGNOSTIC CONTROL switch to OFF.
- Press SYSTEM RESET.

Altering Main Storage:

- Set up address as you would in displaying main storage.
- Set the half-word of data to be entered into the data keys.
- Press STORE.

Displaying a General Purpose Register (GPR):

- Set STORAGE SELECT switch to GP.

- Enter into the storage address bank of keys labeled REGISTER SELECT the number in binary of the register desired, and the specific half-word desired in the portion marked HALF-WORD.
- Press DISPLAY. The half-word will be displayed in the storage data lights.

Displaying a Floating Point Register (FPR):

- Set STORAGE SELECT switch to FP.
- Put the register number in the REGISTER SELECT keys and the half-word desired in the HALF-WORD portion.
- Press DISPLAY. The half-word will be displayed in the storage data lights.

Altering Registers:

- Follow same procedure as in displaying a register.
- Put data in storage data keys.
- Press STORE.

Stopping on Address Compare:

- Put the address you wish to stop on in the storage address keys.
- Turn ADDRESS COMPARE to MS STOP.

Altering the Current PSW:

- Proceed as in displaying the current PSW.
- Put new data in the data keys.
- Hit STORE and the new data will be entered.

Clearing Main Storage:

- Press STOP.
- Hit SYSTEM RESET.
- Set IC to zero.
- Set RATE switch to SINGLE CYCLE.
- Put 0200 in storage data keys.
- Flip ROS REPEAT INSN down.
- Hit START.
- Set RATE switch to PROCESS.
- Flip up all data keys.
- Flip up ROS REPEAT INSN.
- Hit START.
- Hit SYSTEM RESET.
- Hit STORE.

Altering Main Storage:

- Machine must be in manual mode.
- Set address to be stored into, in the address keys.
- Set STORAGE SELECT switch to MAIN.
- Put data to be stored in the data keys.
- Press STORE.

Note: Any location not ending in 0, 4, 8, or C, must be stored at its proper byte location in the selected word.

Displaying Local Storage:

There are four "sectors" in local storage:

- 00 - Channel Sector
- 01 - Working Sector
- 10 - Floating Point Registers (FPR)
- 11 - General Purpose Registers (GPR)

The machine must be in manual mode in order to display any sector of local storage.

- Put sector number to be displayed in address keys 22 and 23.

- Put word to be displayed in address keys 24-27.
- Set STORAGE SELECT switch to LOCAL.
- Press display switch. The display will appear in the L register.

Altering Local Storage:

- Machine must be in manual mode.
- Put sector number in address keys 22 and 23.
- Put word to be stored into in address keys 24-27.
- Set STORAGE SELECT to LOCAL.
- Put data in the data keys.
- Press STORE.

Stopping on Address Compare:

Do not perform the following steps while a program is executing.

- Put address to be stopped on in the address keys.
- Set IAR switch to STOP.
- The instruction pointed to by the address keys will be executed and the next address in the program will be displayed in the IAR.
- If the instruction pointed to by the address keys is a branch instruction and the branch is taken, the address of the branch instruction will be displayed in the IAR.

Altering the Current PSW:

- Be sure machine is in manual mode.
- Set address keys to 170.
- Set storage switch to LOCAL.
- Put new data in the data keys.
- Hit STORE and the new data will be entered.

MODEL 65

Clearing Main Storage:

- Press STOP.
- Flip data keys up.
- Flip down address keys 0, 21, 22.
- Hit SYSTEM RESET.
- Hit ROS TRANSFER.
- Set STORAGE SELECT to LOCAL and then back to MAIN.
- Hit SYSTEM RESET.

Altering Main Storage:

To alter a doubleword:

- Put address to be altered in the address keys.
- Set STORAGE SELECT to MAIN.
- Put data in the 64 data keys.
- Press STORE.

To alter a byte do the same as above but:

- Put into address keys 21-23 the byte number of the data keys that are to be stored.
- Set STORAGE SELECT to MAIN BYTE.
- Press STORE.

Displaying a General Purpose Register (GPR):

- Set STORAGE SELECT to LOCAL.

- Put GPR number in address keys 20-23.
- Hit DISPLAY. The contents of the register will be displayed in the T register - position 3 of roller 2.

Displaying a Floating Point Register (FPR):

- Set STORAGE SELECT to LOCAL.
- Flip down bit position 19 in the address keys.
- Put FPR number in address keys 21-23.
- Hit DISPLAY; the word pointed to by the address keys is displayed in the T register (roller 2, position 3).

Altering Registers:

Proceed as in displaying a register but:

- Put in the bottom bank of data keys the data you wish to store.
- Hit STORE.

Stopping on Address Compare:

- Put the address to stop on in the address keys.
- Flip down the ADDRESS COMPARE STOP switch.
- Press START.

Altering the Current PSW:

- Put new PSW doubleword in the data keys.
- Store the data at location zero.
- Hit PSW RESTART.

Clearing Main Storage:

- Press STOP.
- Press SYSTEM RESET.
- Set general purpose register 0 to zeros.
- Set PSW to zeros.
- Set the data switches to 90 in first byte, 01 in second.
- Flip down ENABLE STORAGE RIPPLE.
- Flip up all other switches.
- Press LOAD A-B REGS.
- Hit START.
- Flip up ENABLE STORAGE RIPPLE.
- Hit SYSTEM RESET.

Altering Main Storage:

- Press STOP.
- Set address switches to the location of the lowest byte of the doubleword to be altered.
- Set STORAGE SELECT to MAIN STOR.
- Press DISPLAY. Contents of doubleword will be displayed in J register.
- Set data switches to match J register.
- Set new information in the appropriate data switches.
- Press STORE.
- Repeat first four steps to check for accuracy.

Displaying a General Purpose Register (GPR):

- Set STORAGE SELECT to GP REGS.

- Put the register number in the REGISTER SELECT keys.
- Hit DISPLAY. The left word of RBL REG will show the contents of the GPR. The right word will show the contents of the next-higher-numbered register.

Displaying a Floating Point Register (FPR):

- Set STORAGE SELECT to FLP REGS.
- Put the register number in the REGISTER SELECT keys.
- Hit DISPLAY. The FPR will be displayed in RBL REG.

Altering Registers:

To alter a GPR proceed as in displaying and:

- Put data into the left half of the data keys.
- Hit STORE.

To alter a FPR proceed as in displaying and:

- Put data into the data keys. Put the characteristic in the 0 byte; when you display the FPR, the characteristic will be in its proper place.
- Hit STORE.

Stopping on Address Compare:

- Put address to be stopped on in the address keys.
- Flip down the ADDRESS COMPARE STOP switch.
- Press START.

Altering The Current PSW:

- Put new PSW doubleword in the data keys.
- Hit SET PSW. The whole PSW, including the instruction counter, will be set to that data.

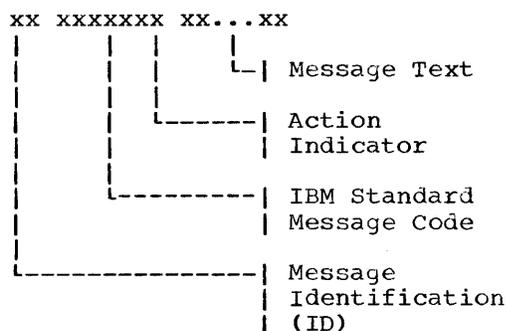


This section describes the general format of console messages written by IBM programs running under the System/360 Operating System.

The exact formats and meanings of individual messages are contained in the publication IBM System/360 Operating System: Messages, Completion Codes, and Storage Dumps.

MESSAGE FORMAT

A console message from an IBM program is up to 120 characters long and has the following format:



Message Identification (ID)

A 2-character numeric field called the message ID appears to the left of, and is separated by one blank from, any message that requires a reply. Use this number with the REPLY command to enable the system to identify the reply. When the message does not require a reply, the message ID does not appear.

The maximum number of IDs, not exceeding 100, is set at the time the system is generated. The system assigns a unique ID to each reply-requesting message.

After processing a REPLY, the ID may be reassigned to another message. If a job ends before you can REPLY to a message, another message is written listing all the message IDs you should ignore.

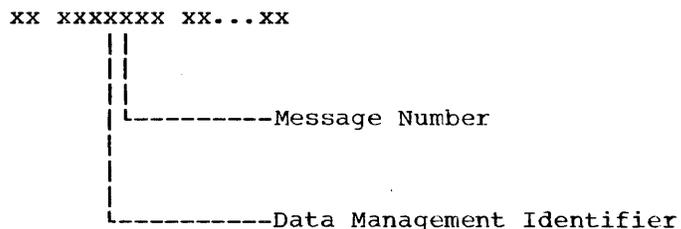
IBM Standard Message Code

The IBM standard message code is a unique alphanumeric code of at least six characters assigned to each message written by IBM programs. The code identifies the program that is the source of the message.

For example, codes starting with the following characters identify the programs listed:

<u>Code</u>	<u>Program</u>
IBC	Independent Utilities
IEA	Supervisor
IEB	Utilities
IEC	Data Management
IEE	Master Scheduler
IEF	Job Scheduler
IEG	TESTRAN
IEH	System and Support Utilities
IEI	System Generator
IEJ	FORTRAN Compiler (E)
IEK	FORTRAN Compiler (H)
IEL	PL/I Compiler (E)
IEM	PL/I Compiler (F)
IEP	COBOL Compiler (E)
IEQ	COBOL Compiler (F)
IER	Sort/Merge
IES	Report Program Generator
IET	Assembler (E)
IEU	Assembler (F)
IEW	Loader
IEX	ALGOL Compiler
IEY	FORTRAN (G)
IFB	System Environment Recording: SER0, SER1, OBR, SDR
IFC	Environment Recording Edit and Print
IFD	Online Test Executive
IFF	Graphic Programming Support
IGE	I/O Error Routines
IHC	Library Subroutines (FORTRAN)
IHD	Library Subroutines (COBOL)
IHE	Library Subroutines (PL/I)
IHF	Library Subroutines (Common)
IHG	Update Analysis
IHI	Library Subroutines (ALGOL)

Data Management Messages: Characters 4, 5, and 6 of the 6-character IBM standard message code are further defined for data management as follows:



The data management identifier is used to relate the message text to a specific data management function. The values the identifier can have, and their associated functions are:

- 0 End of Volume
- 1 Open
- 2 Close
- 3 Data Set Security
- 6 Direct Access Device  
Space Management (DADSM)
- 7 Tape Label Creation

Sort/Merge Messages: Messages beginning with IER (Sort/Merge messages) are not written to the console unless such action was explicitly requested at system generation time. These messages are for information only and require no operator response. They are documented in the publication IBM System/360 Operating System: Sort/Merge, Form C28-6543.

### Action Indicator

The action indicator is a 1-character alphabetic mnemonic that denotes the general class of action required. It appears immediately following the IBM standard message code. It is separated from the message text by at least one blank character.

The following action indicators are defined for System/360 Operating System programs:

- A: An "await action" condition that requires you to perform a specific action before the task will continue. The action is specified either in the message text field or in a supplemental listing. An example of such action is the mounting of required data volumes.
- D: An "await decision" condition that requires you to choose between alternative courses of action. For example, if a job requires more input/output devices than are currently available,

you can either vary a device or cancel the job.

I: An information message that does not require immediate action. For example, this type of message could be used to indicate that a certain control program phase has been completed, or that certain tape units are now available for the mounting of new volumes.

W: A "cannot proceed" condition that stops processing until the action to be taken is determined, and this action is performed. For example, this condition could be caused by a suspected machine malfunction requiring investigation by the Customer Engineer.

S: A "cannot proceed" condition caused by a hardware malfunction that cannot be corrected by retry. Run the standard, stand-alone diagnostic program called SEREP, following the procedure given in Chapter 4.

### Message Text

The message text is a brief statement of information, a request for action, or a request by the program for information to be supplied.

Internal Serial Numbers: You may find internally-generated serial numbers of the form LGLxxx or /xxxxx in the text of messages requesting you to dismount unlabeled tape volumes. If you are asked to retain these volumes, mark them with their internal serial numbers, using any convenient means, such as masking tape and a felt-tipped pen. The system may later ask you to remount the volumes, using their internal serial numbers, in an order different from the one in which they were dismounted.

Not all of the terms listed here are used in this book, but sooner or later you will run across all of them, and be able to come back and look them up.

access method: A technique for moving data between main storage and an input/output device.

address constant: A number, or a symbol representing a number, used in calculating storage addresses.

alias: Another name for a member of a partitioned data set; another entry point of a program.

allocate: Assign a resource to a job or task.

asynchronous: Without regular time relationship; unexpected or unpredictable with respect to the execution of a program's instructions.

attach (task): To create a task control block and present it to the supervisor.

attribute: A trait; for example, attributes of data include record length, record format, data set name, associated device type and volume identification, use, creation date, etc.

auxiliary storage: Data storage other than main storage.

basic access method: Any access method in which each input/output statement causes a machine input/output operation to occur. (The primary macro instructions used are READ and WRITE.)

batch processing: (See stacked job processing.)

block (records): 1. To group records to conserve storage space or to increase the efficiency of access or processing. 2. A blocked record. 3. A portion of a telecommunications message defined as a unit of data transmission.

block loading: Bringing the control section of a load module into adjoining positions of main storage.

buffer (program input/output): A portion of main storage into which data is read, or from which it is written.

catalog: 1. The collection of all data set indexes maintained by data management. 2. To include the volume identification of a data set in the catalog.

cataloged data set: A data set that is represented in an index or series of indexes.

cataloged procedure: A set of job control statements in the SYS1.PROCLIB data set. The procedure can be used by naming it in an execute (EXEC) statement.

CAW (channel address word): A word in main storage at location 72 that specifies the location in main storage where a channel program begins.

CCW (channel command word): A double word at the location in main storage specified by the CAW. One or more CCWs make up the channel program that directs channel operations.

channel: A hardware device that connects the CPU and main storage with the I/O control units.

concatenated data set: A group of logically connected data sets.

control block: A storage area used by the operating system to hold control information.

control dictionary: The external symbol dictionary and relocation dictionary, collectively, of an object or load module.

control program: The routines in the operating system that manage resources, implement data organization and communications conventions, or contain privileged operations.

control section: That part of a program specified by the programmer to be a relocatable unit, all of which is to be loaded into adjoining main storage locations.

control volume: A volume that contains one or more indexes of the catalog.

CPU (central processing unit): The unit of a system that contains the circuits that control and perform the execution of instructions.

CSW (channel status word): A word in main storage at location 64 that provides information about the termination of an input/output operation.

data control block: A control block used by access routines in storing and retrieving data.

data definition name (ddname): A name appearing in the data control block of a program which corresponds to the name field of a data definition statement.

data definition (DD) statement: A job control statement that describes a data set associated with a particular job step.

data management: Those parts of the control program that provide access to data sets, enforce data storage conventions, and regulate the use of input/output devices.

data organization: The arrangement of a data set.

data set: The major unit of data storage and retrieval in the operating system, consisting of a collection of data in one of several prescribed arrangements and described by control information that the system has access to.

data set control block (DSCB): A data set label for a data set in direct-access storage.

data set label (DSL): A collection of information that describes the attributes of a data set, and that is normally stored with the data set; a general term for data set control blocks and tape data set labels.

deferred entry: An entry into a subroutine that occurs as a result of a deferred exit from the program that passed control to it.

deferred exit: The passing of control to a subroutine at a time determined by an asynchronous event rather than at a predictable time.

device independence: The ability to request input/output operations without regard to the characteristics of the input/output devices.

device name: Usually, the general name for a kind of device, specified at the time the system is generated. For example, 2311 or 2400 or TAPE. (See unit name.)

direct access: Retrieval or storage of data by a reference to its location on a volume, rather than relative to the previously retrieved or stored data.

dispatching priority: A number assigned to tasks to determine the order in which they will use the central processing unit in a multitask situation.

dump (main storage): 1. To copy the contents of all or part of main storage onto an output device, so that it can be examined. 2. The data resulting from 1. 3. A routine that will accomplish 1.

entry point: Any location in a program to which control can be passed by another program.

event: An occurrence of significance to a task; typically, the completion of an asynchronous operation, such as input/output.

event control block (ECB): A control block used to represent the status of an event.

exchange buffering: A technique using data chaining to avoid moving data in main storage, in which control of buffer segments and user program work areas is passed between data management and the user program.

exclusive segments: Segments in the same region of an overlay program, neither of which is in the path of the other. They cannot be in main storage simultaneously.

execute (EXEC) statement: A job control statement that designates a job step by identifying the load module to be fetched and executed.

extent: The physical locations on input/output devices occupied by or reserved for a particular data set.

external reference: A reference to a symbol defined in another module.

external symbol: A control section name, entry point name, or external reference; a symbol contained in the external symbol dictionary.

external symbol dictionary (ESD): Control information associated with an object or load module which identifies the external symbols in the module.

fetch (program): 1. To load requested load modules into main storage, relocating them as necessary. 2. A control routine that accomplishes 1.

F format: A data set record format in which the logical records are the same length.

generation data group: A collection of successive, historically related data sets.

inclusive segments: Overlay segments in the same region that can be in main storage simultaneously.

index (data management): 1. A table in the catalog structure used to locate data sets. 2. A table used to locate the records of an indexed sequential data set.

initial program loading (IPL): The initialization procedure which loads the nucleus and begins normal operations.

initiator: The part of the job scheduler that selects jobs and job steps to be executed, allocates input/output devices for them, places them under task control, and at completion of the job, supplies control information for writing job output on a system output unit.

input stream: Job control statements entering the system; may also include input data.

input queue: A queue of job information the job scheduler uses to select the jobs and job steps to be processed.

installation: A particular computing system, in terms of the overall work it does and the people who manage it, operate it, apply it to problems, service it, and use the results it produces.

job: A unit of work for the system from the stand-point of installation accounting and control. A job consists of one or more job steps.

job control statement: A control statement in the input stream that identifies a job or defines its needs.

job library: A set of user-identified partitioned data sets used as the main source of load modules for a given job.

job management: A general term for the work done by the job scheduler and master scheduler.

job queue: (See input queue.)

job scheduler: The part of the control program that controls input streams and system output, obtains input/output resources for jobs and job steps, attaches tasks corresponding to job steps, and otherwise regulates the use of the system by jobs. (See reader, initiator, output writer.)

job (JOB) statement: The control statement in the input stream that identifies the beginning of a series of job control statements for a single job.

job step: A unit of work associated with one processing program or one cataloged procedure, and related data.

language translator: Any assembler, compiler, or other routine that accepts statements in one language and produces equivalent statements in another language.

library: 1. A collection of objects (for example, data sets, volumes, card decks) associated with a particular use, and identified in a directory. See job library, link library, system library. 2. Any partitioned data set.

limit priority: A number associated with a task in a multitask operation, representing the highest dispatching priority that the task can assign to itself or to any of its subtasks.

link library: A partitioned data set which, unless otherwise specified, is used in fetching load modules referred to in execute (EXEC) statements and in ATTACH, LINK, LOAD, and XCTL macro instructions.

linkage: The way two routines or modules communicate.

linkage editor: A program that produces a load module by changing object modules into a form acceptable to fetch, combining object modules and load modules into a single new load module, resolving symbolic cross references among them, replacing, deleting, and adding control sections automatically on request, and providing overlay facilities for modules requesting them.

load: To read a load module into main storage.

load module: The output of the linkage editor; a program in a form suitable for loading into main storage for execution.

locate mode: A way of providing data by pointing to its location instead of moving it.

logical record: A record that is defined in terms of the information it contains rather than by its physical traits.

macro instruction: A macro instruction statement, the corresponding macro instruction definition, the resulting assembler language statements, and the machine language instructions and other data produced from the assembler language statements; loosely, any one of these representations of a machine language instruction sequence.

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

master scheduler: The part of the control program that responds to operator commands and returns required information.

MFT: Multiprogramming with a fixed number of tasks.

module (programming): A program unit that is input to, or output from, a single execution of an assembler, compiler, or linkage editor; a source, object, or load module.

move mode: A way of providing data by moving it instead of pointing to its location.

multijob operation: Concurrent execution of job steps from two or more jobs.

multiprogramming: Using main storage for more than one program at a time.

multitask operation: Multiprogramming; called multitask operation to express parallel processing not only of more than one program, but also of a single reenterable program used to do many tasks.

MVT: Multiprogramming with a variable number of tasks.

name: A set of one or more characters that identifies a statement, data set, module, etc., and that is usually associated with the location of that which it identifies.

nucleus: That part of the control program that must always be present in main storage. Also, the main storage area used by the nucleus and other transient control program routines.

object module: The output of a single execution of an assembler or compiler, which constitutes input to linkage editor. An object module consists of one or more control sections in relocatable, though not executable, form and an associated control dictionary.

operator command: A statement to the control program, issued via a console device, which causes the control program to provide requested information, alter normal operations, initiate new operations, or terminate existing operations.

output queue: A collective term for the 36 queues of control information describing system output data sets. These queues specify to an output writer the location

and disposition of system output. Each of the 36 queues is associated with one output class.

output writer: A part of the job scheduler that writes output data sets onto a system output unit, independently of the program that produced such data sets.

overlay: To place a load module or a segment of a load module into main storage locations occupied by another load module or segment.

overlay (load) module: A load module that has been divided into overlay segments, and has been provided by linkage editor with information that enables overlay supervisor to load the desired segments when requested.

overlay segment: (See segment.)

overlay supervisor: A control routine that controls fetching of overlay segments on the basis of information recorded in an overlay module by linkage editor.

parallel processing: Concurrent execution of one or more programs.

partition: In systems with MFT, an area of main storage set aside for a job.

partitioned data set: A data set divided into several members. Each member has a unique name and is listed in a directory at the beginning of the data set. Members can be added or deleted as needed. Records within members are organized sequentially.

path: A series of segments that form the shortest distance in a region between a given segment and the root segment.

PCP (primary control program): The basic control program, which provides sequential scheduling with no multiprogramming.

physical record: A record that is defined in terms of physical qualities rather than by the information it contains.

polling: A technique by which each of the terminals sharing a communications line is periodically checked to determine if it requires servicing.

post: Note the occurrence of an event.

private library (of a job step): A partitioned data set other than the link library or the job library.

problem program: A routine that solves problems, monitors industrial processes, sorts and merges records, performs computations, processes transactions against stored records, etc., as opposed to a control program or a language translator.

processing program: Any program, such as a problem program or a language translator, other than a control program.

PSW (program status word): A double word in main storage used to control the order in which instructions are executed, and to hold and indicate the status of the system in relation to a particular program.

qualified name: A data set name that is composed of multiple names separated by periods (for example, TREE.FRUIT.APPLE).

qualifier: All names in a qualified name other than the rightmost, which is called the simple name.

queued access method: An access method that automatically governs the movement of data between the program using the access method and input/output devices. (The primary macro instructions used are GET and PUT.)

reader: The part of the job scheduler that analyzes an input stream.

ready condition: The condition of a task that is ready to be performed by the central processing unit.

real time (interval timer): Actual time.

record: A unit of data.

reenterable: A program concurrently usable by two or more tasks.

region: In systems with MVT, an area of main storage set aside for a job step task or a system task.

relocation: The changes of address constants required when a change of origin of a module or control section is made.

relocation dictionary: That part of an object or load module which identifies all relocatable address constants in the module.

resource: Any facility of the system required by a job or task, including main storage, input/output devices, the central processing unit, data sets, and control and processing programs.

resource manager: Any control program routine responsible for the handling of a resource.

return code: A number placed in a designated register (the "return code register") at the completion of a program. The number is established by user-convention and may be used to influence the execution of succeeding programs or, in the case of an abnormal end of task, it may simply be printed for programmer analysis.

return code register: A register in which a user-specified condition code is placed at the completion of a program.

reusable: Usable by two or more tasks. (See reenterable, serially reusable.)

root segment: The first segment in an overlay program.

scatter loading: Placing the control sections of a load module into non-adjointing positions of main storage.

scheduler: (See master scheduler and job scheduler.)

secondary storage: Auxiliary storage.

seek: Position the access mechanism of a direct-access device at a specified location.

segment: 1. The smallest unit (one or more control sections) that can be loaded during execution of an overlay program. 2. As applied to telecommunications, a portion of a message that can be contained in a buffer.

serially reusable: Usable by another task after the current use has been concluded.

short block: A block of F format data which contains fewer logical records than are standard for a block.

simple buffering: A technique for controlling buffers in such a way that the buffers are assigned to a single data control block.

simple name: The rightmost component of a qualified name (for example, APPLE is the simple name in TREE.FRUIT.APPLE).

source module: A series of statements which make up the entire input to a single execution of an assembler or compiler.

stacked job processing: A technique that permits multiple job definitions to be grouped (stacked) for presentation to the system, which automatically recognizes the jobs, one after the other.

storage block: An area of main storage consisting of 2048 bytes to which a storage key can be assigned.

subtask: A task that is created by another task by means of the ATTACH macro instruction.

supervisor: The key controlling part of the operating system. The supervisor governs the use of the main frame - the central processing unit and main storage.

SVC (supervisor call): An instruction which causes an SVC interruption in the hardware to give control to a control program routine (called an SVC routine) for some specific action, such as reassigning parts of main storage or retrieving data from an I/O device.

synchronous: Occurring with a regular or predictable time relationship.

SYSIN: A system input stream. Also, a name used as the data definition name of a data set in the input stream.

SYSOUT: A system output stream. Also, an indicator used in data definition statements to signify that a data set is to be written on a system output unit.

system input unit: A device specified as a source of an input stream.

system library: The collection of all cataloged data sets at an installation.

system macro instruction: A macro instruction that provides access to operating system facilities.

system output unit: An output device shared by all jobs.

system residence volume: The volume on which important operating system routines and data are located.

SYSCTLG: An optional system data set on the primary system residence device containing addresses relating installation data set names to specific volume numbers.

SYS1.GENLIB: A data set, normally kept offline, used for system generation.

SYS1.LINKLIB: A system data set containing the system program modules that are not either permanently resident in main storage or resident in the SYS1.SVCLIB.

SYS1.LOGREC: A system data set on the primary system residence device containing information regarding system failures.

SYS1.MODLIB: A data set, normally kept offline, used for storing program modules during modification by processes such as linkage editing and system generation.

SYS1.NUCLEUS: A system data set on the primary system residence device containing the IPL program and the primary nucleus.

SYS1.PROCLIB: A data set containing cataloged procedures -- handy sets of control statements that can be called into use by EXEC statements.

SYS1.SVCLIB: A system data set on the primary system residence device containing all of those SVC routines, I/O error recovery routines, and access method routines, that are not permanently resident in main storage.

SYS1.SYSJOBQE: A system data set used by the scheduler as a storage and work area for information about the input and output streams. Contains the input queue and output queue.

task: A unit of work for the central processing unit from the standpoint of the control program; the basic multiprogramming unit under the control program.

task control block (TCB): The consolidation of control information related to a task.

task dispatcher: The control program routine that selects from the task queue the task that is to be performed by the central processing unit.

task management: The work done by the supervisor; regulating the use of the central processing unit and resources other than input/output devices.

task queue: A queue of all the task control blocks present in the system at any one time.

telecommunications: Data transmission between a system and remotely located devices via a unit that performs format conversion and controls the rate of transmission.

teleprocessing: A term associated with IBM telecommunications equipment and systems.

test translator: A facility that allows various debugging procedures to be specified in assembler language programs.

text: The control sections of an object or load module.

throughput: The rate at which work can be handled by a system.

transmittal mode: The way the contents of an input buffer are made available to the program, and the way a program makes records available for output.

turnaround time: The time between submission of a job to a computing center and the return of results.

U format: A data set format in which blocks are of unknown length.

unit name: Usually, the unit address of a particular device, specified at the time a system is installed. For example, 191 or 293. (See device name.)

user: Anyone who requires the services of a computing system.

V format: A data set format in which logical records are of varying length and include a length indicator; and in which V format logical records may be blocked, with each block containing a block length indicator.

volume: That part of a unit of storage media which is accessible to a single read/write mechanism.

volume table of contents (VTOC): A table associated with a direct-access volume, which describes each data set on the volume.

wait condition: The condition of a task that needs one or more events to occur before the task can be ready to be performed by the central processing unit.

wait state: The state of the system when no instructions are being processed, but the system is not fully stopped. The system can accept I/O and external interruptions, and can be put through the IPL procedure.



- Abbreviations, MVT command 42
- Action indicator 82
- Active volumes, where to mount, MVT 43
- Allocation, device
  - MFT 26,27
  - MVT 40,42,43
  - PCP 12-14
- Altering main storage
  - Model 30 75
  - Model 40 76
  - Model 50 77
  - Model 65 78
  - Model 75 79
- Alternate coding key 9,21,35,57
- Alternate console shift 57
- Automatic commands
  - MFT 24
  - MVT 38
  - PCP 11
- Automatic volume recognition (AVR) 13,27
  
- Bld1 option 57
  
- Cancel command
  - MFT 28
  - MVT 44
  - PCP 14
- CAW (channel address word) 65
- CCW (channel command word) 65
- Class, output 39
- Clearing main storage
  - Model 30 75
  - Model 40 76
  - Model 50 77
  - Model 65 78
  - Model 75 79
- Command format conventions
  - MFT 28
  - MVT 44
  - PCP 14
- Command statement 73
- Commands in the input stream 57
- Commands on a composite console 57
- Commands on a 1052 57
- Console malfunction bypass 61
- Control program 7
- Control statements 71
- CSW (channel status word) 67
  
- DCB (data control block) 19,34,54
- DD (data definition) statement 73
- Delimiter statement 73
- Device names, MVT 54
- Direct-access space, freeing of, MVT 43
- Display command
  - MFT 28
  - MVT 45
  - PCP 15
- Displaying main storage 63
- Dump 15,28,44
- Dump/restore program 69
  
- Entering partition size 23
- EOB (end of block) 9,21,35,57
- EOF (end of file) 9,21,37
- Error recovery, MFT 24
- Examples of initial commands
  - MFT 24
  - MVT 38
  - PCP 11
- Exec (execute) statement 73
- External old PSW 65
  
- Glossary 83-89
  
- Halt command, MVT 46
- Hardware debugging aids 69
- Hold command, MVT 46,53
  
- IEBGENER program 54
- IEFPRT program 18,34
- IFCEREPO program 71
- Imperative mount 13,27
- Independent utility programs 69
- Initiator 9,35
- Initiators, number to start, MVT 38
- Input
  - MFT 25
  - MVT 39
  - PCP 12
- I/O command analysis 67
- I/O old PSW 65
- Internal serial numbers 82
- Interruptions 7
- IPL (initial program loading)
  - MFT 21
  - MVT 35
  - PCP 9
  
- Job control through Hold and Release, MVT 53
- Job extraction from tape input stream, MVT 53
- Job queue parameter specification, MVT 52
- Job statement 73
  
- Log command, MVT 46
- Log, system 40
  
- Machine check old PSW 65
- Main storage display 63
- Message
  - format 81
  - ID 81
  - text 82
- MFT (multiprogramming with a fixed number of tasks) 21-34
- Min option 59
- Modify command, MVT 46
- Mount command
  - MFT 29
  - MVT 47
  - PCP 15

MVT (multiprogramming with a variable number of tasks) 35-54

NIP (nucleus initialization program) 11,23,37

Nucleus

- MFT 23
- MVT 37
- PCP 11

Null statement 73

OBR (outboard recorder) 71

Operator commands

- MFT 27-32
- MVT 43-51
- PCP 14-17

Output

- MFT 26
- MVT 39
- PCP 12

Partition 21

Partition planning 25

Partition redefinition 23

PCP (primary control program) 9-20

Permanent main storage assignments 63

Priority

- MFT 21
- MVT 35

Program old PSW 65

PSW (program status word) 59,65

PSW, altering current

- Model 30 75
- Model 40 76
- Model 50 77
- Model 65 78
- Model 75 79

PSW display 61

Qbf option 59

Ram option 57

Reader

- MFT 25
- MVT 35
- PCP 9

Reader device names, MVT 54

Reduced system activity, causes, MVT 53

Region 45

Register display and alteration

- Model 30 75
- Model 40 76
- Model 50 77
- Model 65 78
- Model 75 79

Release command, MVT 47,53

Reply command

- MFT 29
- MVT 48
- PCP 15

Request command

- MFT 29
- PCP 16

Reset command, MVT 48

Restarting

- MFT 25
- MVT 38
- PCP 12

Rq option 57

Rsvc option 57

Scheduler

- general 7
- MFT 24
- MVT 35
- PCP 9

Scheduler shifting 34

SDR (statistical data recorder) 71

Secondary nucleus loading 59

SER (system environment recording) 69,71

Serial numbers, internal 82

Set command

- MFT 30
- MVT 48
- PCP 16

Shift command, MFT 31

Sort/merge messages 82

Sqs option 57

Standard message codes 81

Start command

- MFT 31
- MVT 49
- PCP 16

Starting

- MFT 21,33
- MVT 37,51
- PCP 9,18

Stop command

- MFT 32
- MVT 50
- PCP 17

Stopping

- MFT 25
- MVT 38
- PCP 12

Stopping on address compare

- Model 30 75
- Model 40 76
- Model 50 77
- Model 65 78
- Model 75 79

Storage use improvement, MVT 38

Supervisor

- general 7
- MFT 21
- MVT 35

SVC (supervisor call) 7

SVC old PSW 65

SYSIN (system input)

- MFT 21
- MVT 39
- PCP 9

SYSOUT (system output)

- MFT 26
- MVT 39
- PCP 12

SYSOUT tape

- MFT 34
- MVT 54
- PCP 18

SYSOUT writers with one initiator, MVT 43

System operation 7

System parameter specification 57

System status determination, MVT 52

SYS1.LOGREC 38

Technical terms 83-89  
Telecommunications 25

UCS (universal character set) procedures,  
MVT 54

UCS utility program 71

Unload command

MFT 32

MVT 51

PCP 17

Updating system data sets

MFT 34

MVT 53

Vary command

MFT 32

MVT 51

PCP 18

Volume label creation 71

Volume mounting

MFT 27

MVT 42

PCP 13

Work volumes, MVT 43

Writelog command, MVT 51

Writer, output

MFT 26

MVT 40

PCP 12

Writer device names, MVT 54

# IBM Technical Newsletter

File Number S360-20  
Re: Form No. C28-6540-5  
This Newsletter No. N28-2273  
Date November 15, 1967  
Previous Newsletter Nos. None

## IBM SYSTEM/360 OPERATING SYSTEM OPERATOR'S GUIDE

This technical newsletter corresponds to Release 14 and contains amendments to the Operator's Guide publication. Replacement and/or supplemental pages to be inserted in the publication are noted below. Corrections and additions to text and/or illustrations are indicated by a vertical bar to the left of the text or illustration and a bullet (•) to the left of an illustration caption.

<u>Pages to be Inserted</u>	<u>Pages to be Removed</u>
Cover, Preface	Cover, Preface
11, 12, 12.1	11, 12
17, 18, 18.1	17, 18
25, 26, 26.1	25, 26
31, 32, 32.1	31, 32
37, 38, 38.1	37, 38
41, 42, 42.1	41, 42
43, 44, 44.1	43, 44
45, 46	45, 46
47, 48	47, 48
49, 50	49, 50
51, 52, 52.1	51, 52
59, 60, 60.1	59, 60

### Summary of Amendments

The amendments to this publication are mainly in support of release 14 of the operating system. These amendments include:

- Information on multiple output classes and scheduler improvements.
- Descriptions of the generalized START and STOP commands.
- How to limit the apparent size of main storage.
- Additional information on restarting the system, and on DISPLAY A.
- Minor corrections.

Note: Please file this cover letter at the back of the publication. Cover letters provide a quick reference to changes and a means of checking receipt of all amendments.

IBM Corporation, Programming Systems Publications, P.O. Box 390, Poughkeepsie, N.Y. 12602

# IBM Technical Newsletter

File Number S360-20  
Re: Form No. C28-6540-5  
This Newsletter No. N28-2306  
Date November 15, 1967  
Previous Newsletter Nos. N28-2273

## IBM SYSTEM/360 OPERATING SYSTEM OPERATOR'S GUIDE

This technical newsletter corresponds to Release 14 and contains amendments to the Operator's Guide publication. Replacement pages to be inserted in the publication are noted below. Additions to text are indicated by a vertical bar to the left of the text.

Pages to be  
Inserted

49,50

Pages to be  
Removed

49,50

### Summary of Amendments

The amendments to this publication support Disk SYSIN in release 14 of the operating system.

Note: Please file this cover letter at the back of the publication. Cover letters provide a quick reference to changes and a means of checking receipt of all amendments.

**READER'S COMMENT FORM**

IBM System /360 Operating System  
Operator's Guide

Form C28-6540-5

- Is the material:

	Yes	No
Easy to read? .....	<input type="checkbox"/>	<input type="checkbox"/>
Well organized? .....	<input type="checkbox"/>	<input type="checkbox"/>
Complete? .....	<input type="checkbox"/>	<input type="checkbox"/>
Well illustrated? .....	<input type="checkbox"/>	<input type="checkbox"/>
Accurate? .....	<input type="checkbox"/>	<input type="checkbox"/>
Suitable for its intended audience? .....	<input type="checkbox"/>	<input type="checkbox"/>
  
- How did you use this publication?

<input type="checkbox"/> As an introduction to the subject	Other .....
<input type="checkbox"/> For additional knowledge	
  
- Please check the items that describe your position:

<input type="checkbox"/> Customer personnel	<input type="checkbox"/> Operator	<input type="checkbox"/> Sales Representative
<input type="checkbox"/> IBM personnel	<input type="checkbox"/> Programmer	<input type="checkbox"/> Systems Engineer
<input type="checkbox"/> Manager	<input type="checkbox"/> Customer Engineer	<input type="checkbox"/> Trainee
<input type="checkbox"/> Systems Analyst	<input type="checkbox"/> Instructor	Other .....
  
- Please check specific criticism(s), give page number(s), and explain below:

<input type="checkbox"/> Clarification on page(s) .....	<input type="checkbox"/> Deletion on page(s) .....
<input type="checkbox"/> Addition on page(s) .....	<input type="checkbox"/> Error on page(s) .....

Explanation:

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

**YOUR COMMENTS PLEASE . . .**

This publication is one of a series which serves as reference for systems analysts, programmers and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

Fold

Fold

FIRST CLASS  
PERMIT NO. 81  
POUGHKEEPSIE, N.Y.

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



POSTAGE WILL BE PAID BY  
  
IBM Corporation  
P.O. Box 390  
Poughkeepsie, N.Y. 12602

Attention: Programming Systems Publications  
Department D58

Fold

Fold



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

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

**IBM**<sup>®</sup>

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

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