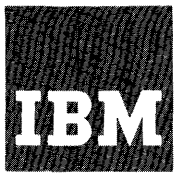
File No. S360-20
Form C28-6540-7

OS



Systems Reference Library

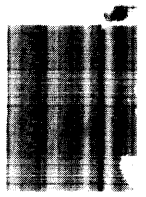
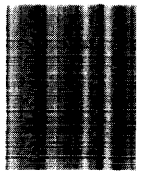
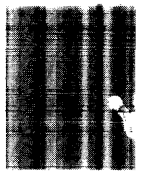
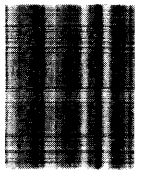
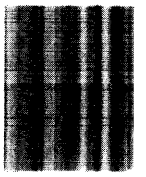
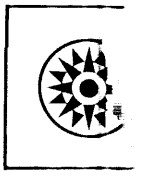
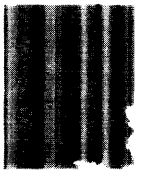
IBM System/360 Operating System

Operator's Guide

This book 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). The remote job entry facility is described. Instructions are given telling:

- How to start, stop, and restart the operating 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 run the operating system, you must have the required book listed below, as well as this guide. In addition, you are encouraged to read the recommended books.

REQUIRED BOOK

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

RECOMMENDED BOOKS

IBM System/360 Operating System:

Job Control Language, Form C28-6539

Job Control Language Charts, Form C28-6632

Utilities, Form C28-6586

HOW TO USE THIS GUIDE

- If this is your first reading, read the Introduction, the chapters describing the type of system you have at your installation, and then Chapters 5, 6, and 7.
- 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 7. 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.

Eighth Edition (May 1968)

This edition, Form C28-6540-7, applies to release 15/16 of IBM System/360 Operating System and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. This edition obsoletes its predecessor, Form C28-6540-6, and the publication IBM System/360 Operating System: Using the IBM 2250 Display Unit Model 1 as System Operator's Console, Form C27-6941. Besides correcting errata, this edition contains additions and changes supporting release 15/16:

- New chapter on MFT (multiprogramming with a fixed number of tasks)
- New chapter on RJE (remote job entry)
- New section on how to use the IBM 2250 as an operator's console (MVT)
- New information on the "job classes" incremental improvement (MFT and MVT), time-slicing (MFT and MVT), shared DASD (PCP, MFT, and MVT), and the graphic job processor (MFT and MVT)
- New hints on using the universal character set feature and on general operating techniques.

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

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

Figures

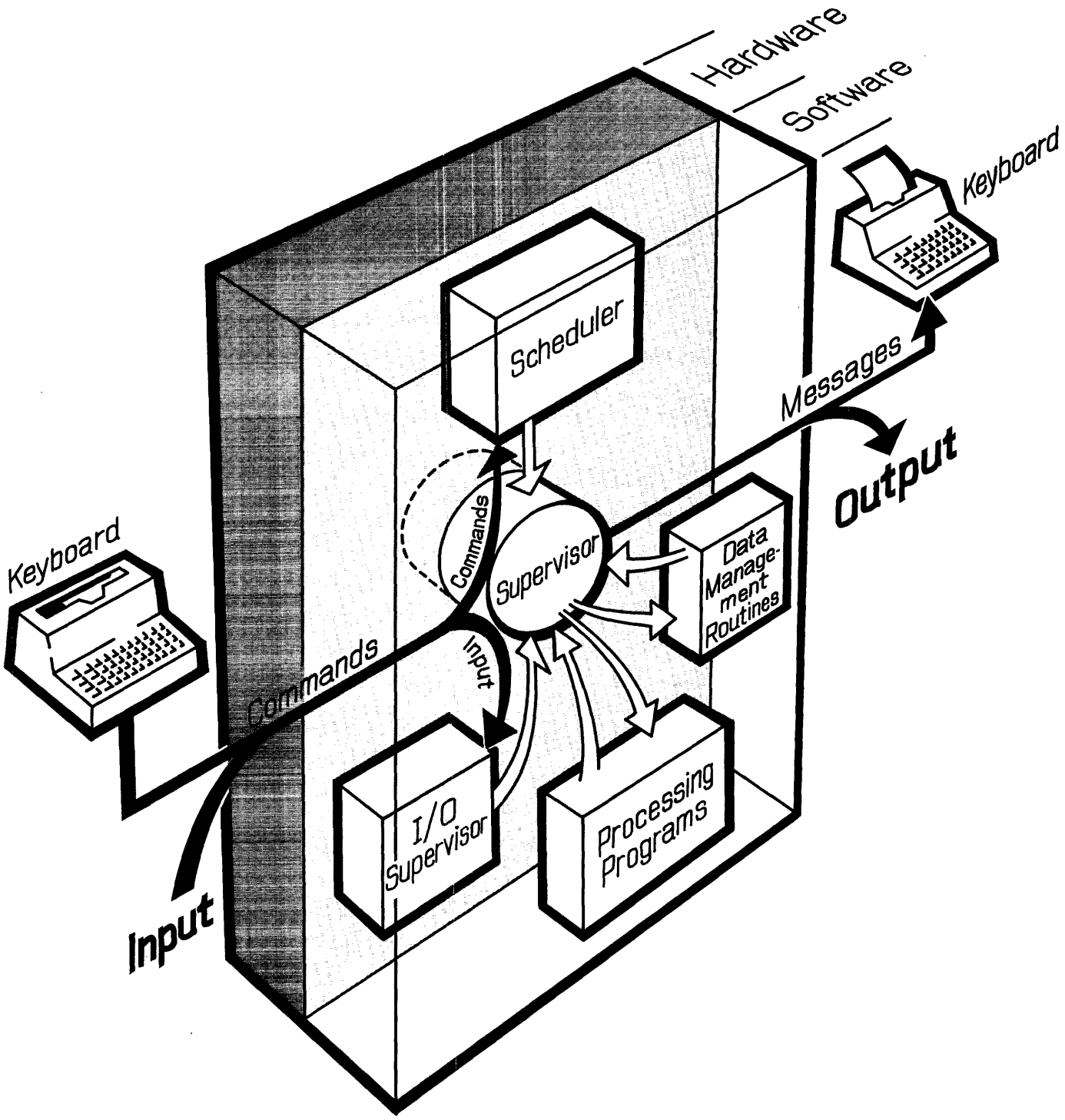


Figure 1. How the System Works

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 hit 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 among 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).

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.

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 hitting 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 5.

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, hit 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 holding down the alternate coding key and hitting the numeric 5 key.

You can cancel a typed line by holding down the alternate coding key and hitting 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) hit the card reader's STOP key, place the command in the reader, and hit the START and EOF keys.

You can also enter most commands through a SYSIN (system input) stream by using command statements.

Each command statement, or group of command statements, must precede a JCB, an EXEC, or a null statement.

Commands are accepted as soon as they are read.

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.

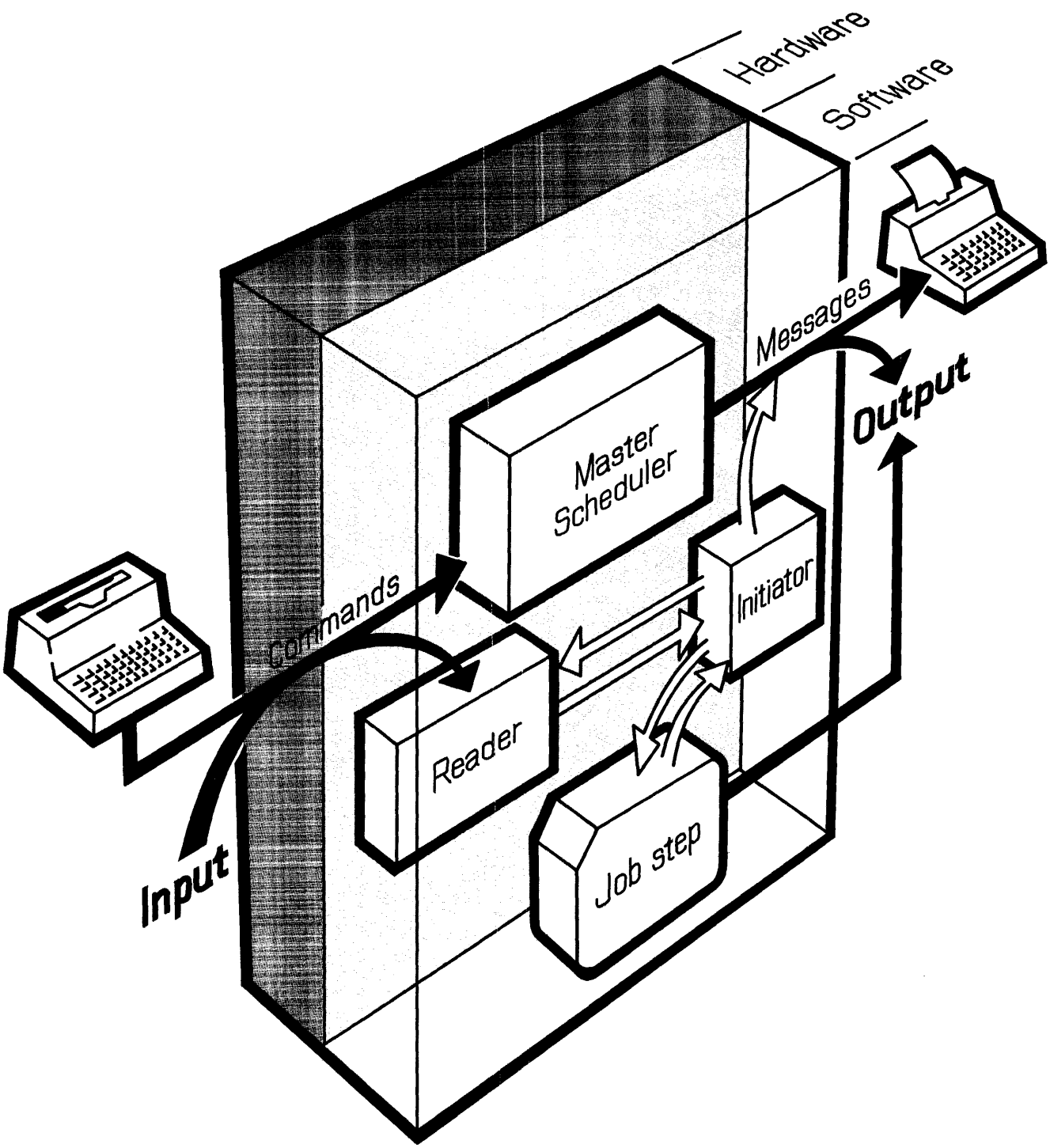


Figure 2. How the Scheduler Works in Systems With the Primary Control Program

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 hit the LOAD key on the panel.

Hitting 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 5.

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 12 bits 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 and Codes.

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'll 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 5, under the heading "How to Specify System Parameters."

If no changes are to be made, issue REPLY id,'U', or simply signal EOB.

After NIP completes its preparation of the system, it passes control to the master scheduler. You'll 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 low-order 12 bits 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 don't 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 demount 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 hit the START key on the device to continue processing.

Never mount a blank tape volume, unless specifically directed to do so.

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 standard labels on 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 demounted, the system unloads the devices, if possible, and writes you messages identifying the volumes being unloaded.

When you receive a mounting message for a 2321 data cell, hit 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 don't have to hit RESET -- closing the door performs the same function in this case.

Occasionally you may be asked by your installation not to mount volumes on certain devices, and not to make those devices ready, because you are running a version of the operating system that was generated for a slightly different set of devices.

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 demounted 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 alphameric characters may be defined by your installation, but these are for use by your programmers in their data definition statements.

Don't use symbolic names in operator commands.

Shared DASD Option: If your installation is using a system with this option, be sure to read the section "How to Use the Shared DASD Option" in Chapter 5.

OPERATOR COMMANDS

This section contains a description of the commands you use to control the operating system. For convenience, the commands are given in alphabetical order.

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

*CANCEL	C	*SET	T
DISPLAY	D	START	S
MOUNT	M	STOP	P
*REPLY	R	UNLOAD	U
+*REQ		VARY	V

* These commands cannot be entered in the input stream.

\+ The REQ command cannot be abbreviated.

Console Commands, other than SET and START, are accepted whenever you issue them.

To issue a SET or START command, you must get a READY message.

A REQ command will give you a READY message. It 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.

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.

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. This command cannot be entered into the input stream.

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.

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.

Operation	Operand
{CANCEL / C}	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.

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'll receive a message even if you have not used the DISPLAY command.)

This 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. In addition, this command provides you with the names of non-temporary data sets in mount and K (keep) -type dismount messages.

Operation	Operand
{DISPLAY / D}	{JOBNAMES [,T] / STATUS / DSNAME}

JOBNAMES [,T]
specifies that the name of each job is to be displayed both when the job starts and when it ends, and that unit record allocation is to be displayed when the step starts.

If the T parameter is used in conjunction with the JOBNAMES parameter, the system displays the time of the day in addition to the jobnames. The time is shown in the format

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

In systems without the timer option, this parameter is ignored.

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.

DSNAME
specifies that the system is to display, within the mount and K (keep) type dismount messages, the name of the first non-temporary data set allocated to the volume to which the messages refer.

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 demountings 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 / M}	unitname [,VOL=(NI,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. 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.

VOL=(NL,serial)
specifies that the volume does not have standard labels. The alphameric 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.

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. This command cannot be entered into the input stream.

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 / R}	id,'text'

id
specifies the 2-character message identification field of the message requesting the reply. This field is described in Chapter 6.

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. This command cannot be entered into the input stream.

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.

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. This command is also used to specify the device on which the accounting data set (SYS1.ACCT) resides. Any combination of the above may be specified.

The SET command cannot be entered into the input stream.

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 don't include the timer option, the CLOCK parameter is ignored.

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

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) or (unitname,F) or (,F)
specifies (1) the name of the direct access device (other than an IBM 2321) on which the volume containing the input work queue (SYS1.SYSJOBQE) resides or (2) that the system is to format the input work queue, or both. This parameter is used only in the first SET command after IPL.

Space for the input work queue must have already been allocated on the volume which is mounted on the specified device.

You need not specify a 3-character unit name if one of the following conditions exists:

- the SYS1.SYSJOBQE data set is cataloged,
- the SYS1.SYSJOBQE data set is contained on the system residence volume.

If the system is to format the input work queue prior to the first job initiation, you must specify ,F following or without the 3-character unit name. For example, Q=(unitname,F) specifies that the system is to format the input work queue on the volume residing on the direct access device referred to by unitname; Q=(,F) specifies that the system is to format the

input work queue either on the volume to which the SYS1.SYSJOBQE data set is cataloged or on the system residence volume.

If this parameter is not given, the system searches the system catalog (SYSCTLG) to determine and select the volume that contains the SYS1.SYSJOBQE data set. If this data set is not cataloged, the system assumes that the data set is contained on the system residence volume.

The system issues an error message when either of the following conditions exists:

1. The volume to which the SYS1.SYSJOBQE data set is cataloged is not mounted.
2. The system cannot locate the SYS1.SYSJOBQE data set on the selected volume.

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 searches the system catalog (SYSCTLG) to determine and select the volume that contains the SYS1.PROCLIB data set. If this data set is not cataloged, the system assumes that the data set is contained on the system residence volume.

The system issues an error message if either of the following conditions exists:

1. The volume to which the SYS1.PROCLIB data set is cataloged is not mounted.
2. The system cannot locate the SYS1.PROCLIB data set on the selected volume.

ACCT=([unitname][,N])
specifies the name of the direct access device on which the SYS1.ACCT data set resides; it further specifies that accounting data is to be written starting at the beginning of the SYS1.ACCT extent.

If you omit unitname, the system assumes that the SYS1.ACCT data set is contained on the system residence volume.

If you omit N, the system attempts to write accounting data from the end of the last accounting record written into the SYS1.ACCT data set.

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 / S}	[{RDR / WTR},unitname [,volumeserial] [,parmvalue] [,DSN=datasetname] [,FILESEQ=filesequencenumber]]

Note: If you do not use the volumeserial parameter, you must indicate its absence by a comma; example:
START WTR,282,,A,DSN=YEARTODATE

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 / P}	{JOBNAMES / WTR,unitname / STATUS / DSNAME}

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.

DSNAME
specifies that the system is to stop the display of the names of non-temporary data sets as initiated by the DSNAME parameter of the DISPLAY command. For more information, see the discussion of the DISPLAY command.

UNLCAD -- Prepare Volume for Demounting

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

The UNLCAD command causes a volume on an input/output device to be prepared for demounting.

When the volume is ready to be demounted, you'll receive a message. (The message may not be received until the current job is completed.)

Operation	Operand
{UNLCAD / U}	unitname

unitname
specifies the unit address of the input/output device to be prepared for demounting. 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 / V}	unitname {,ONLINE / ,OFFLINE}

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

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 by a subsequent VARY command or, if an appropriate system message is received, by issuing REPLY id,'unitname'.

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 5, which describes general techniques.

HOW TO START A SYSTEM WITH THE PRIMARY CONTROL PROGRAM

1. Set the LOAD UNIT switches to the channel, control unit, and device to address the system residence volume.
2. Hit 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 hit the LOAD key again.
3. Specify System Parameters. 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 5.
4. 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 low-order 12 bits 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.
5. After you receive a READY message, enter a SET command with the parameters your installation requires.
6. 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
7. Enter a START command with no parameters as your last command.

HOW TC DETERMINE SYSTEM STATUS

If the system goes into an enabled wait state, first check the console sheet for unsatisfied mount requests or other messages and reply appropriately. In the case of unsatisfied mount requests, the proper volumes should be mounted promptly or the job canceled.

If the system is in a wait state, check whether the last message on the console requires a reply and if so, reply appropriately.

If there are no such outstanding messages on the console, it is necessary to CANCEL the job currently being processed. If a DISPLAY JOBNAMES is in effect, the jobname of the last job started is the one to CANCEL.

Such an enabled wait state may be caused by one of the following:

- Unsatisfied mount requests or error recovery messages.
- A programming error.
- In a Shared DASD system (see Chapter 5) where two or more systems are sharing the same device, an interlock due to a programming error. This situation is usually characterized by all CPUs being in an enabled wait state. In this case, it may be necessary to CANCEL a job in only one of the systems to resolve the interlock. If not all CPUs are in an enabled wait state, proceed with caution; a system could simply be waiting for a device to be released.

HOW TO PRINT A SYSOUT TAPE

In systems with the primary control program, execute the program called IEFPRT 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 1: 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.

Note 2: If your PRINTAP DD statement has been coded incorrectly or is not included in your JCL deck, the system cannot start the output printer. Therefore, if your output printer does not start although it is ready, check your JCL deck for a properly coded PRINTAP DD statement. The system places a return code of 8 in register 15 whenever it is unable to start the output printer for this reason.

In systems with MFT, main storage beyond the nucleus area is separated into from 1 to 52 areas called partitions.

These areas may consist of as many as 15 problem program partitions, as many as 3 resident input reader partitions, and as many as 36 resident output writer partitions, providing the total number of partitions does not exceed 52.

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 or while the system is running. See Figure 3.

Jobs, read in by system input readers, are placed in input work queues according to CLASS and PRTY parameters, specified on the JOB statement. MFT schedules jobs independently for each partition.

Each partition accepts jobs from as many as three input work queues.

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.

When P0 enters a wait state, P1 is given control. If P0 and P1 are both in a wait state, P2 is given control. In a system which contains 15 problem program partitions, P14 is not given control unless all of the higher priority partitions are unable to execute.

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 hitting 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 5.

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, hit 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 holding down the alternate coding key and hitting the numeric 5 key.

You can cancel a typed line by holding down the alternate coding key and hitting 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) hit the card reader's STOP key, place the command in the reader, and hit the START and EOF keys.

You can also enter most commands through a SYSIN (system input) stream by using command statements.

Each command statement, or group of command statements, must precede a JCB, an EXEC, or a null statement.

Commands are accepted as soon as they are read - they aren't 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.

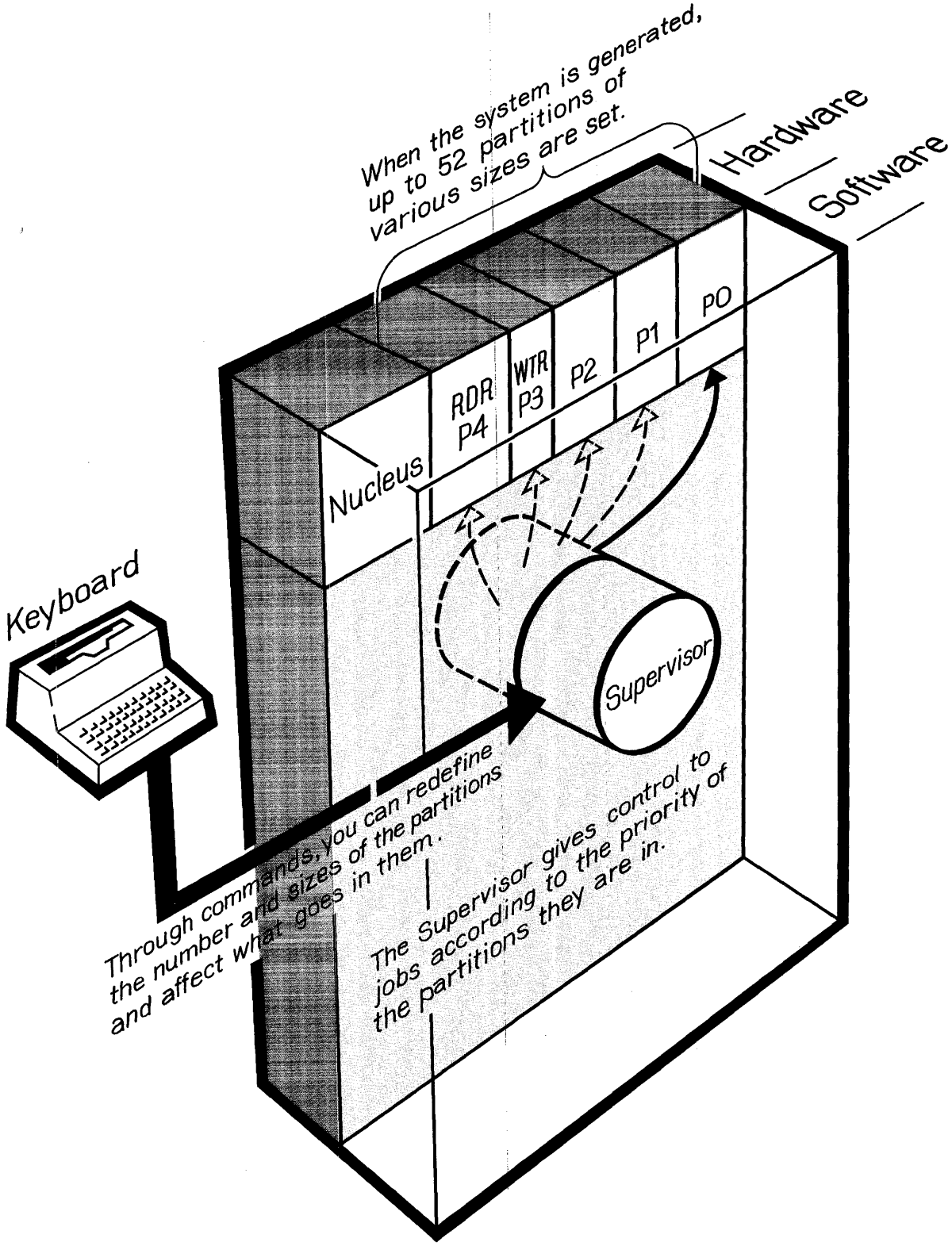


Figure 3. How Main Storage is Partitioned in Systems With MFT

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 hit the LOAD key on the panel.

Hitting 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 5.

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 12 bits 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 and Codes.

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'll 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, the RSVC option, or the SQS option. Explanations for the various options, and instructions on how to alter the options, are given in Chapter 5, under the heading "How to Specify System Parameters."

If no changes are to be made, issue REPLY,id,'U', or simply signal EOB.

After NIP completes its preparation of the system, it passes control to the master scheduler. You'll 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'll 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 low-order 12 bits of the current PSW as the system waits.

Redefining Partitions

When redefining partitions, you can redefine the number, size, and job classes of partitions. If the time-slicing option was selected when the system was generated, you can also redefine their time-slicing specifications.

The number of partitions can be equal to or less than that specified at system generation.

The total size of all partitions and the nucleus cannot exceed the amount of main storage available.

Partition Redefinition at System Initialization: The master scheduler 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, the master scheduler issues the message:

```
id IEE801D CHANGE PARTITIONS?
```

If partitions are to be redefined and a list of the current partition definitions is desired, enter:

```
REPLY id,'YES,LIST'
```

If partitions are not to be redefined, enter NO in place of YES. Omit characters ,LIST if a list is not desired. Generally

use LIST even when no redefinition is intended. This causes a list of the current partition definitions and a list of the job classes currently being serviced to be recorded on the console device.

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 the message:

id IEE802A ENTER DEFINITION

(preceded by the current definitions if LIST was specified).

You can now redefine the partitions. See "Entering the Partition Definitions" for an explanation of the redefinition procedure.

Partition Redefinition After System Initialization: To redefine partitions after system initialization enter:

DEFINE LIST

Omit LIST from the DEFINE command if no list is desired. In response to DEFINE, the system issues

id IEE802A ENTER DEFINITION.

Enter the new partition definitions at this point.

Entering the Partition Definitions: When responding to the message ENTER DEFINITION or CONTINUE DEFINITION you can change a partition's size or job classes, specify the last partition which is to be active, and request a list of all current partition definitions including all job classes which are currently being serviced. You also can end all further definition in the present series, and can cancel all definitions which have been entered since the last ENTER DEFINITION message. Examples are given following.

1. To change the size of a partition, enter:

   REPLY id,'Pn=size'

   where n is the partition number and size is the new partition size expressed as an absolute decimal value (e.g., 40960), or as a multiple of 1024 plus the suffix K (e.g., 40K). The minimum acceptable size is 8192, or 8K. If the system has the optional storage protection feature, sizes that aren't multiples of 2K will be increased to the next multiple of 2K. A size of zero causes a partition to remain inactive.

   Note: If you attempt to decrease the size of a single partition, without a corresponding increase to an adjacent partition, the message IEE812I Pn HAS xxxx EXCESS BYTES ADDED is issued because the excess storage is always added by the system to the lowest priority partition among those being defined. Therefore the partition size is not decreased.

2. To change the job class(es) of a partition enter:

   REPLY id,'Pn=class'

   where n is the partition number (0-51) and one of the following is substituted for class:

   xxx
   one to three alphabetic characters from A through O. Each character identifies a job class that the partition is to service. The first character identifies the primary job class, the second the secondary job class, and the third the tertiary job class.

   Jobs will be selected for the partition from the primary job class as long as there are entries in that job class queue. If there are no entries in that job class queue, the scheduler attempts to select a job from the secondary class assignment. If there are no entries in the secondary class queue, the tertiary queue is checked. If there are no entries in any of the job class queues serviced by the partition, the partition waits until a job is enqueued for one of its classes.

   RDR
   a resident reader partition.

   WTR
   a resident writer partition.

3. To specify the last partition to be active, enter:

   REPLY id,'Pn=LAST'.

   All partitions with partition numbers higher than the partition defined as LAST are considered inactive. If LAST is entered in error, the partition definition can be reentered, omitting LAST. LAST cannot be specified for more than one partition.

   If 'Pn=LAST' refers to a partition that has previously been made inac-

tive, partition P(n-1) will be the last partition because Pn doesn't have a size associated with it.

4. To change the size and job classes of a partition enter:

   REPLY id,'Pn=(size,class)' or

   REPLY id,'Pn=(class,size)'.

   Size, class, and LAST can be used singly or in any combination of two or three. If used in combination they must be separated by commas and enclosed in parentheses.

5. To request a list of current partition definitions, enter:

   REPLY id,'LIST'.

   At system initialization, all job classes being serviced by active partitions are also listed.

6. To request a list of all job classes being serviced by active partitions, enter:

   REPLY id,'CLASS'.

   At system initialization, this parameter is unnecessary if LIST is specified.

7. To end all redefinitions in the current series (a series begins with the reply following the message ENTER DEFINITION and ends with the last reply in which END is specified) and to allow the system to implement all replies just entered, you should enter:

   REPLY id,'END'.

   During a series, you can enter a definition for each partition number as frequently as you wish. At the end of the series, the most recent entry for each partition number is accepted by the system.

8. To cancel all partition definitions in a series, enter:

   REPLY id,'CANCEL'.

   The partition definitions remain as they were before the current series of definitions was entered. The current series is ended.

9. To change the partitions in a time-sliced group, enter:

   REPLY id,'TMSL=(Pi-Pj)'

   where i and j are contiguous partition numbers (0 through 51) which define the range of partitions that are to belong to the group. The total number of partitions to be time-sliced cannot exceed the number specified at system generation time.

10. To change the time slice, enter:

    REPLY id,'TMSL=value'

    where the value given for the time slice must be 20 or larger, up to four digits, representing the maximum number of milliseconds each task in the time-sliced group can have control of the CPU at any one time.

11. To change the time slice and the partitions in the time-sliced group, enter:

    REPLY id,'TMSL=(Pi-Pj,value)'

    Where i, j, and value have the meanings given in examples 9 and 10.

12. To cancel time slicing, enter:

    REPLY id,'TMSL=CANCEL'

13. To get a console listing of the current time-slicing specification, enter:

    REPLY id,'TMSL'

14. Replies can specify more than one request. Partitions can be redefined, LIST, TMSL, CLASS, END, CANCEL, and LAST can all be specified in a single reply. Any combination of these optional parameters can be specified in any order so long as they are separated by commas. The only restriction is that commands cannot occupy more than one line. A command entered on the console cannot exceed 128 characters; the two examples below are on more than one line only because of the narrow column width used in this book.

    Example 1:

    ```
    REPLY id,'LIST,P1=(36K,RDR),
    P2=(ABC,40K),TMSL=(P1-P2,200),
    CLASS,P0=0,P3=(LAST,WTR),P1=30K,
    END'
    ```

    In this example, a list of partition definitions is requested. The list will include the new definition for P0 through P3 as defined in this reply. Partition 1 is initially redefined as a resident reader partition of 36864 bytes. The last entry before END

again changes the size of this reader partition to 30720 bytes. Partition 2 is redefined as a 40960 byte partition having a primary job class A, secondary job class B, and tertiary job class C.

The time-sliced group will now be made up of partitions 1 and 2, and the time slice will be 200 milliseconds. A list of the job classes currently being serviced by active partitions is requested. Job classes being defined in this reply are included. Partition 0 is to be inactive. Partition 3 is redefined as a resident writer partition. Any partitions numbered higher than 3 are to be inactive. The end of the current series of redefinitions is specified; all redefinitions entered since the last ENTER DEFINITION message are now implemented by the system.

Example 2:

```
REPLY id,'P0=(12K,JFE),P1=(18K,MO),
END'
```

This example reactivates partition 0 which was made inactive in Example 1. The reply defines a size of 12288 bytes for this partition and assigns job classes J, F, and E to it. The reply reduces the size of partition 1 from 30720 bytes to 18432 bytes and assigns job classes M and O to it. The reply shown in this example doesn't affect partitions 2 and 3.

Error Recovery: In systems with MFT, you can cancel errors made before an entry is complete by holding down the alternate coding key and hitting the numeric 0 key. Errors in redefinition recognized before the END entry are canceled simply by entering the correct definition.

Three types of errors can be recognized by the system after each reply. The system issues the following messages:

1. DEFINITION PARAMETER ERROR, REPLY AGAIN if a syntax error has been detected,

2. Pn NOT DEFINABLE, REPLY AGAIN if the partition designated by n was not defined at system generation,

3. DEFINITION DELIMITER ERROR, REPLY AGAIN if a delimiter error has been detected.

When you receive one of the above messages, the system ignores the information you furnished with the reply to which the message relates.

Four additional types of errors can be recognized by the system after END is specified. The system issues the following messages:

1. TOTAL SIZE OF PARTITIONS IS xxxxx BYTES TOO LARGE FOR STORAGE where xxxxx is the amount by which storage is exceeded,

2. PARTITIONS DEFINED EXCEED AVAILABLE SPACE if redefined adjacent partitions exceed the space originally specified for the previously defined partitions;

3. PROBLEM PROGRAM PARTITIONS EXCEED 15, RESPECIFY if more than 15 problem program partitions have been specified;

4. CHANGED PARTITIONS NOT ADJACENT, RESPECIFY if the operator attempted to redefine nonadjacent partitions.

When one of the above messages is issued, the system does not consider the series ended. All syntactically correct definitions in replies entered since the ENTER DEFINITION message are considered still active. Only the END parameter in the replies preceding these messages is ignored by the system. See the Messages and Codes book for an explanation of the action required to recover these errors.

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 can 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.

You will receive a SPECIFY JOB QUEUE PARAMETERS message after you issue the SET command with the formatting parameter (Q=(unit name,F)). Instructions telling how to answer this message are given under the heading "How to Specify Job Queue Parameters" in Chapter 5.

After issuing SET, you can enter any commands in any order. START reader, START writer, and START INIT must be issued if they have not been specified by your installation at system generation to be started automatically.

If your installation has already specified automatic START commands, I/O devices are automatically allocated to an input reader and an output writer.

An initiator is started in all scheduler-size partitions, and the commands are written on the console as if you had keyed them in yourself.

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

If your installation has specified automatic START commands, the commands will appear on the console in the following format (assuming that the input device is a card reader with the unit address 00C and the output device is a printer with the unit address 00E):

```
START WTR.Pn,00E *

START RDR.S,00C *

START INIT.ALL *
```

The automatic START command for the system output writer will have the partition number of the highest priority partition which is defined as a writer, placed in the command (Pn will be replaced with the partition number by the master scheduler).

If you don't want to start a writer in that partition, you must override the command and then issue a START command specifying the partition number in which you want the writer started. See Example 4.

The automatic START command for the system input reader will start a system-assigned transient reader in the first available (highest priority) problem program partition.

If your system configuration includes a partition which has been defined as a resident reader partition, you will probably want to override the automatic command and issue a START command for the resident reader partition (the command specifying a transient reader will not start a reader in the resident reader partition). See Example 5 for the command sequence to start a resident reader.

The automatic START command for the initiator will cause an initiator to be started in all large problem program partitions.

If you want to start the initiator only in selected partitions, override this automatic command and issue a START INIT command for each selected partition.

The DISPLAY command with the JOBNAMES 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 JOBNAMES, you'll have the job names available if you have to cancel any jobs.

Examples:

1. To start a system that has automatic START reader, START writer, and START INIT commands (the CLOCK operand in each example is optional):

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

2. To start a system that has automatic commands, but to suppress them:

```
SET DATE=yy.ddd,CLOCK=hh.mm.ss,
   Q=(unitname,F),AUTO=NONE
DISPLAY JOBNAMES
START writer.identifier
START reader.identifier
START INIT.P2,,,ABC
```

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

```
SET DATE=yy.ddd,CLOCK=hh.mm.ss,
   Q=(unitname,F)
VARY unitname,OFFLINE
START reader.identifier
START INIT.P0,,,A
START INIT.P2
```

4. To start a system that has automatic START WTR, START RDR and START INIT commands, but to start a system output writer in a lower priority partition than that specified in the command,

```
SET DATE=yy.ddd,CLOCK=hh.mm.ss,
    Q=(unitname,F),AUTO=NYY
START WTR.P8,00E
```

5. To start a system that has automatic START WTR, START RDR, and START INIT commands, but to start a resident

reader in partition 0, and start initiators in partitions 1, 3, and 5 only.

```
SET DATE=yyy.ddd CLOCK=hh.mm.ss,
    Q=(unitname,F),AUTO=YNN
START RDR.P0,00C
START INIT.P1
START INIT.P3
START INIT.P5
```

Planning the Work for each Partition

Through careful classification of jobs, you can achieve optimum balance between your resource requirements and the utilization of those resources.

The CLASS parameter on the JOB statement can inform the system of the non-conflicting nature of jobs, permitting them to be scheduled for concurrent execution.

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 STOP GFX for the graphics interface task (if the task was initiated).

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

To restart the system after a failure, follow the same steps you took in starting the system, but when entering the SET command omit the "F" suffix from the "Q=unitname" parameter or omit the "Q=unitname" parameter entirely.

To restart the system enter the command (CLOCK is optional):

```
SET DATE=yy.ddd,CLOCK=hh.mm.ss
```

By omitting the "Q=unitname,F" parameter, job queue data set information is saved.

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'll 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's 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 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.

In systems with MFT, as many as three input readers can bring jobs into the system concurrently. Each input reader is assigned its own SYSIN stream. You issue a START command to start a new input reader, and a STOP command to stop a reader. (Input readers also 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.

Job Classes

Jobs can be grouped, depending upon your installation needs, into as many as 15 different job classes (A-O).

Each class has a one-character classname and includes all job control information for input jobs associated with the classname. This permits grouping of jobs with a common characteristic. The characteristic might be a job with much input/output, a job requiring a large amount of CPU processing, or possibly one which requires special control volumes.

The CLASS for a job is specified by the CLASS parameter of the JOB statement.

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 transmittal to the device specified by the programmer for its output. See Figure 4.

Output Classes

Your installation, depending on its needs, can group SYSOUT data into as many as 36 different output classes.

Each class has a one-character classname and includes all system messages and SYSOUT data sets the system associates with the classname, allowing 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 output classes 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 printed 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 output class specified by the MSGCLASS= parameter of the JOB statement. A SYSOUT data set is in the output 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 MFT. The system messages for this job are in class B, while the output data set is in class C.

```
//EXAMPLE  JOB    ,'AUSER',MSGCLASS=B,      X
//                CLASS=N,PRTY=12
//STEP1    EXEC   PGM=PROGRAM
//RESULT   DD     UNIT=2311,SYSOUT=C,       X
//                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 command.

In systems with MFT, 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 command to start a new output writer, but also must issue a STOP command to stop one.

By using START, you specify the association between the output classes and devices, by naming (a) the device to be allocated to the output writer and (b) the output classes 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 and WTR is the procedurename of a writer.

```
START WTR.P3,00E,,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.P3,185,,C
```

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

A job step writes SYSOUT data sets at high speed onto a direct access device. Later, after the entire job is completed, an output writer will handle the data sets concurrently with other jobs.

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.
- Cperator 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 don't 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.

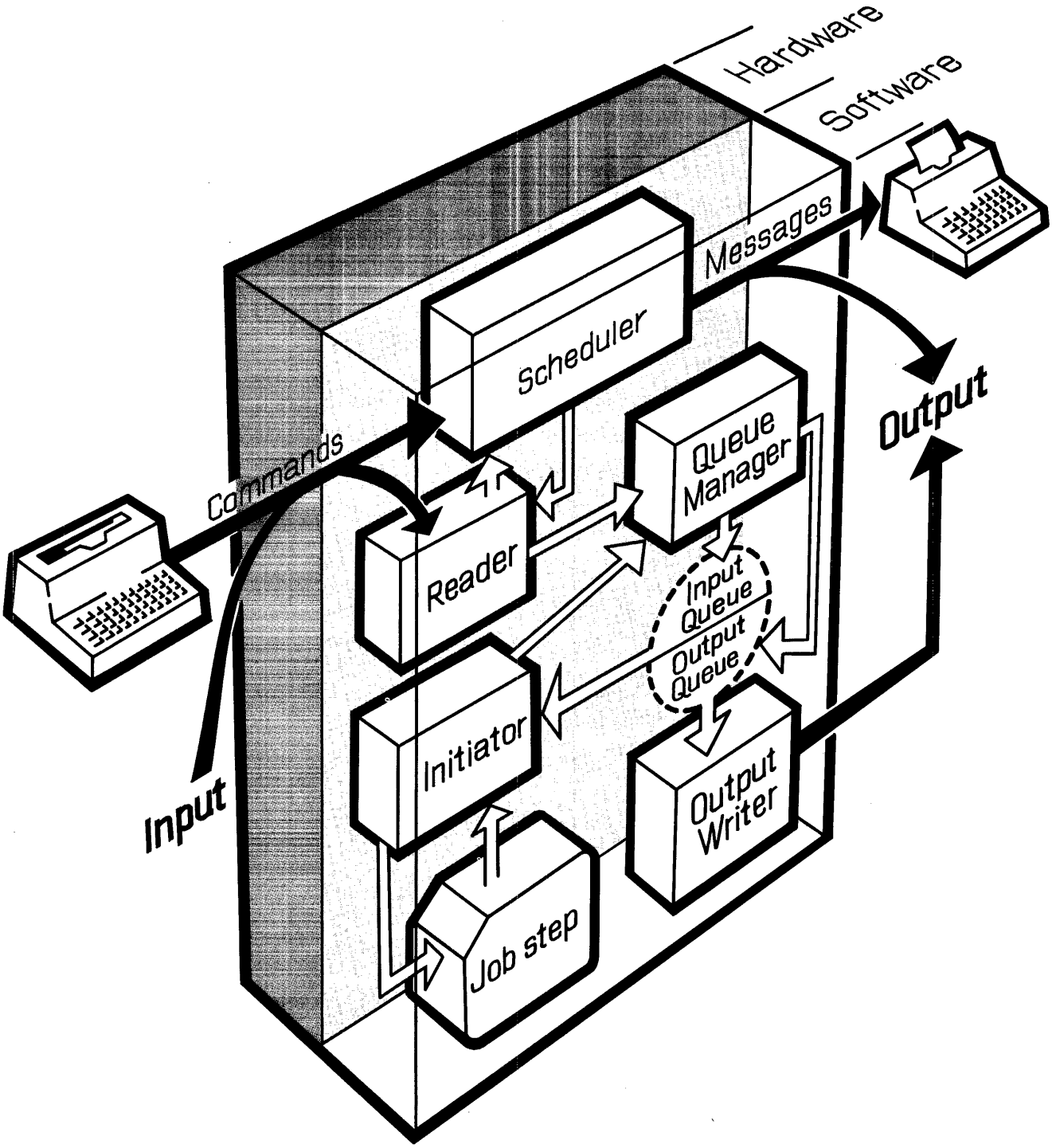


Figure 4. How The Scheduler Works in Systems With MFT

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.

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 demount 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.

Never mount a blank tape volume, unless specifically directed to do so.

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 standard labels on 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.

Occasionally you may be asked by your installation not to mount volumes on certain devices, and not to make those devices ready, because you are running a version of the operating system that was generated for a slightly different set of devices.

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.

Shared DASD Option: If your installation is using a system with this option, be sure to read the section "How to Use the Shared DASD Option" in Chapter 5.

Wait Messages

During operation, you may receive a message informing you that a job is waiting in a partition for resources to become available.

When these resources become available, the job resumes processing. If, however, the job is waiting in the only large partition in the system (or the only large partition not containing an unending job) the needed resources will never become available unless you free the partition for use by the system.

If the reader is waiting for space or if the initiator is waiting for space or a device, reply "CANCEL" in order to free the partition for further work.

If a transient writer is waiting for work, issue a STOP command in order to free the partition. If another large partition is available in the system, no reply is required.

In some cases a message series will be issued indicating that a job cannot be initiated because needed data sets are reserved by another job.

You will have to make a decision, based on your knowledge of the jobs in the system, the system configuration, and the data sets requested.

You will receive message IEF864D REPLY' 'RETRY' or 'CANCEL' as the last message in the series. Your reply to this message will probably be 'RETRY' unless you know that:

1. There is only one scheduler-size partition in the system and therefore no other jobs can be terminated if the scheduler is waiting in that partition.
2. The data sets listed are reserved by an unending job, and therefore will not be likely to be released.
3. The job waiting is being scheduled into a small partition and the scheduler is operating in the only large partition which does not have an unending job in it.

If you reply CANCEL to this message, you can later reenter the job at a time when the data sets will be available.

Your reply to RETRY need not be made immediately; you can wait until other jobs have terminated and freed the necessary data sets.

CPERATCR 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 MFT, abbreviations as well as the full command name can be used when keying in the commands. The usable names and abbreviations are:

CANCEL	C	REPLY	R
*DEFINE	N	RESET	E
DISPLAY	D	SET	T
*HALT	Z	START	S
HCLD	H	STOP	P
MODIFY	F	UNLOAD	U
MOUNT	M	VARY	V
RELEASE	A		

* These commands cannot be entered into the input stream.

When you issue certain commands, such as DISPLAY, you should be aware that confusion may arise due to intermingled messages. While displaying the status of jobs, you may also be informed of system messages, of WTCR messages, or of jobs beginning or ending or both.

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.

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.

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, an abnormal end-of-task storage dump may be requested if the command is received while the job is running.

If the named job is in one of the input work queues 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 its associated 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 for system tasks. The STOP command must be issued at this time.

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.

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

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

If a partition identifier (any one from P0 through P51) of an active partition -- for example, P4 -- is used as a job name, a CANCEL command will cancel both the job named P4 and any system task waiting for allocation in partition 4.

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 this card was omitted, an indicative (partial) dump is taken.

unitname
specifies the name of an I/O device. The system will stop the output writer from completing the writing of the data set being written on the named device. The output writer automatically continues processing further output.

identifier
specifies the identifier of a system task to be terminated during allocation. The identifier is the identifier specified in the START command.

DEFINE - Invoke Dynamic Partition Definition

The DEFINE command is issued after nucleus initialization to allow you to change the size and description of any partition while unaffected partitions continue processing. This command cannot be entered into the input stream.

Operation	Operand
DEFINE / N	[LIST]

LIST
specifies that the current partition definitions will be listed. Job classes associated with currently active partitions and, if time slicing is used, the time-slicing specifications are listed also.

CAUTION: When using the time-slicing option, do not define job classes across the boundaries between a time-sliced group of partitions and partitions that are not time-sliced. For example, don't specify a partition with job classes A,B,C in a time-sliced group, and a partition with job classes C,D,E outside the group. Doing so would allow a job in class C to be put either inside or outside the time-sliced group regardless of the intentions of the programmer of that job.

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.

This command provides the 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.

In addition, the DISPLAY command is used to cause a console display of the time of day and the date or the names of non-temporary data sets in mount and K(keep)-type dismount messages.

Operation	Operand
DISPLAY / D	STATUS JOBNAMES [,T] T A R Q [=list] N [=list] DSNAME jobname

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.

JOBNAMES [,T]
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.

If the T parameter is used in conjunction with the JOBNAME parameter, the system displays the time of the day in addition to the job names. The time is shown in the format:

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

In systems without the timer option, this parameter is ignored.

T
specifies that the time of day and the date are to be displayed in the following format:

```
hh.mm.ss            yy.ddd
|  |  |             |   |
|  |  └---Seconds   |   └---day
|  └------Minutes   └-------year
└---------Hours (00-23)
```

A
specifies that the jobname, current stepname, and partition are to be displayed for each job currently running. If a system task is running, its name and partition will be displayed.

R
specifies that the system is to display;

- The id of each message that required a reply and has not yet been replied to,
- The unit number of each device for which a mount message has been issued but has not been complied with, and
- An indication whether or not any AVR mount messages are pending.

When you use the DISPLAY R command, the system displays the requested information as message IEE110A if any action on your part (reply to messages or mount volumes) is yet required or as message IEE111I if you have complied with all system requests.

Q
specifies that a listing of the number of entries on each of the non-empty input, hold, and output queues is to be displayed.

N
specifies that a listing of jobnames on the hold, input, and output queues is to be displayed.

list
specifies any combination of up to four of the following items:

specific input work queue name (job class A through O)

SOUT (system output queues collectively)

HOLD (system hold queue)

If list includes more than one item, you must separate the specified items by commas and enclose them in parentheses. If no list value is specified, all 15 input work queues, the hold queue, and the output queue are assumed.

DSNAME
specifies that the system is to display, within the mount and K(keep)-type dismount messages, the name of the first non-temporary data set allocated to the volume to which the messages refer.

jobname
specifies that the class, priority, queue location, and position on the queue are to be displayed for the job specified. If STATUS, JOBNAMES, T, A, Q, N, or DSNAME is used as a job name it must be enclosed in parentheses.

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. This command cannot be entered in the input stream.

You must use this command to ensure that important statistics and data records in main storage are not lost permanently.

Operation	Operand
{HALT / Z}	EOD

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 5).

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, a message will be received.

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 specified input work queues 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 that is in a queue which is held because of a HOLD Q command.

Operation	Operand
{HOLD / H}	{jobname / Q [=list]}

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 specified or implied input work queue is to be suspended.

list
specifies any combination of up to four input work queue names (job class A through O). If no list value is specified, all 15 input work queues are assumed. If list includes more than one item, you must separate the specified items by commas and enclose them in parentheses.

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 the operator 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}	[procname.]identifier [,CLASS=classnames] [,PAUSE={FORMS / DATASET}]

procname
specifies the name of a cataloged procedure to start a writer.

identifier
specifies the name of a writer (corresponding to the identifier which was specified in the START command).

CLASS=classnames
specifies one to eight single-character names of the classes to be associated with the output writer; for example, CLASS=ABCD. 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. Note: The system will also accept classnames as a series of characters in parentheses and separated by commas; for example, CLASS=(A,B,C,D). But the form ABCD is preferred to (A,B,C,D).

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 demountings 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
{MCUNT / M}	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.

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.

VOL=(NL,serial)
specifies that the volume does not have standard labels. The alphameric 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.

RELEASE -- Make Job Available for Selection

The RELEASE command is used to resume job selection that has been suspended by the HOLD command.

To release a specific job that was held through a HOLD jobname command, or TYPRUN=HOLD on the JOB card, issue a RELEASE jobname command.

To release jobs held because they were on the specified 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 that is in a queue which is held because of a HOLD Q 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.

Operation	Operand
{RELEASE / A}	{jobname / Q [=list]}

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.

list
: specifies any combination of up to four input work queue names (job class A through C). If no list value is specified, all 15 input work queues are assumed. If list includes more than one item, you must separate the specified items by commas and enclose them in parentheses.

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 a two-digit message identification field of the message requesting the reply. This field is described in Chapter 6.

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 number 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.

Use of the optional CLOCK operand in systems with MFT is suggested if the timer option is included. These systems use the data in this operand when they name system data sets.

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:

```
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) or (unitname,F) or (,F)
specifies (1) the name of the direct access device (other than a 2321) on which the volume containing the input work queue (SYS1.SYSJOBQE) resides or (2) that the system is to format the input work queue, or both. This parameter is used only in the first SET command after IPL.

Space for the input work queue must have already been allocated on the volume which is mounted on the specified device.

You need not specify a 3-character unit name if one of the following conditions exists:

- The SYS1.SYSJOBQE data set is cataloged,
- The volume containing the SYS1.SYSJOBQE data set resides on the device that has been specified at the time of system generation, or
- The SYS1.SYSJOBQE data set is contained on the system residence volume.

If the system is to format the input work queue prior to the first job initiation, you must specify ,F following or without the 3-character unit name. For example, Q=(unitname,F) specifies that the system is to format the input work queue on the volume residing on the direct access device referred to by unitname; Q=(,F) specifies that the system is to format the input work queue either on the volume to which the SYS1.SYSJOBQE data set is cataloged, or on the volume that resides on the device specified at the time of system generation, or on the system residence volume.

If this parameter is not given, the system searches the system catalog (SYSCTLG) to determine and select the volume that contains the SYS1.SYSJOBQE data set. If this data set is not cataloged, the system uses the values provided at the time of system generation, if any such values were provided. If no such values were provided, the system assumes that the data set is contained on the system residence volume.

The system issues an error message when either of the following conditions exists:

1. The volume to which the SYS1.SYSJOBQE data set is cataloged is not mounted.
2. The system cannot locate the SYS1.SYSJOBQE data set on the selected volume.

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 searches the system catalog (SYSCTLG) to determine and select the volume that contains the SYS1.PROCLIB data set. If this data set is not cataloged, the system uses the values provided at the time of system generation, if any such values were provided. If no such values were provided, the system assumes that the data set is contained on the system residence volume.

The system issues an error message if either of the following conditions exists:

1. The volume to which the SYS1.PROCLIB data set is cataloged is not mounted.

2. The system cannot locate the SYS1.PROCLIB data set on the selected volume.

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 RDR, S WTR, 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, readers, writers, and the graphics interface task.

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

procname
specifies the name of the cataloged 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, WTR for an output writer, INIT for an initiator, and GFX for the graphics interface task. For additional information on the graphic job processor, see "How to Specify the Use of the Graphic Job Processor" in the next chapter.

When a START INIT command is entered in a system containing small partitions (any size between 8K and the size of the scheduler), the partition containing the scheduler places the job into the specified (small) partition and initiates the job.

identifier
specifies one of the following:

ALL (used with the procname INIT) to indicate that the initiator is to be started in all partitions, if this is desired.

Pn where n is a partition number (0 through 51) to indicate the partition in which the initiator, graphic interface task, reader, or writer is to be started.

S to indicate that the reader or writer is to be system-assigned. Only one system-assigned reader or writer can be active in a system at any time.

The identifier identifies a reader or writer when multiple readers or writers are started with the same procedure name if either the device name is not specified or it is the same for more than one task, e.g., disk reader.

devicename
specifies the name of the input or output device associated with an input reader or output writer. This can be either a unit name (such as 293) or a general name (such as 2400). If specified, the name will override any corresponding unit specification in the cataloged procedure.

If you don't use this parameter but specify one or both of the parameters volumeserial and parmvalue, you must indicate the absence of the device parameter by a comma. (See the example under "parmvalue" below.)

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. Do not specify a serial number for a direct access volume when starting a writer.

If you don't use this parameter but specify a parmvalue parameter, you must indicate the absence of the volumeserial parameter by a comma. (See the example under "parmvalue.")

parmvalue
specifies parameter values to be passed to the program receiving control as a result of the START command. If parmvalue contains any non-alphameric character, such as an equals sign, parmvalue must be enclosed in parentheses.

The most common parmvalue types are the name of a job in the input stream; the class name parmvalue used by an output writer, for example: S WTR.P1, 00E,,ABC; and the job class identifiers assigned to an initiator.

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 found.

A class name parmvalue may be used when starting an output writer: use this parmvalue to limit the classes which the writer is to process.

The class name parmvalue may be from one to eight characters which represent the classes to be processed. The specified classes are treated on a priority basis where the left-most character indicates the queue to be processed first. A class name parmvalue entry causes all class names specified in the cataloged procedure to be overridden.

The output writer will also accept a class name as a series of characters in parentheses and separated by commas; for example, S WTR.P1,00E,,(A,B,C). But the form ABC is the preferred form, as opposed to (A,B,C).

A START INIT command can specify up to three job class identifiers (A though O). Such a START command must include one comma each for the missing parameters devicename and volumeserial. Example: S INIT.P3,,,MNO. Job class identifiers specified in a START INIT command override job classes previously assigned to a particular partition.

If you specify this parameter but none or only one of the parameters devicename and volumeserial, you must indicate the absence of the latter two by commas. Example: S RDR.Pn,2400,, PAYROLL. The example for a START INIT command given to show the preferred form of specifying class name contains commas for the missing parameters devicename and volumeserial.

gfx=option
: specifies a new value for and overrides the corresponding GFX option value selected at the time of system generation. You can override just one or more or all of the GFX option values specified at the time of system generation. For a description of the gfx=option parameters, see "How to Specify the Use of the Graphic Job Processor" in the next chapter.

  In a START command for a graphics interface task, the gfx=option parameter takes the position of the parmvalue parameter. Therefore, whenever you issue a START command specifying a GFX option value, you must indicate the absence of the parameters devicename and volumeserial by commas. Example: S GFX.Pn,,,(gfx=option, gfx=option).

  You can omit the enclosing parentheses if you specify only one GFX option.

  You cannot specify a keyword-option parameter in a START command for a graphic interface task.

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.)

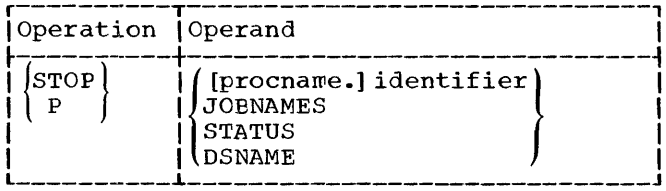
  If such keyword parameters are specified, they will override the corresponding parameters on the DD statement for the input or output device in the cataloged procedure. If the devicename positional parameter is used, the UNIT keyword cannot be used. If the volumeserial positional parameter is used, the VOLUME keyword cannot be used.

  If the input device is a disk, you must use the keyword DSNAME=name 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 VCLUME=SER=volumeid. 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.

STOP -- Stop System Process

The STOP command is used to stop the operation of an input reader, an output writer, an initiator, or the graphics interface task, or to stop a console display of job and data set names.

When you use this command to stop a function other than a display, the procedure does not stop immediately. Instead the system begins the stopping after the procedure finishes handling its current task. After a reader is started, it must process one job before a STOP RDR command will take effect. In addition, input readers stop automatically on meeting an end-of-file condition.



procname
specifies the name of a catalogued procedure defining an input reader, output writer, initiator, or the graphics interface task. For standard procnames, see the discussion of the START command.

identifier
specifies the initiator, graphic interface task, reader or writer to be stopped (corresponding to the identifier specified in the START command).

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

STATUS
specifies that the console display of the data set names (with their volume serial numbers and dispositions of KEEP, CATLG, or UNCATLG) which was initiated by the STATUS parameter of the DISPLAY command, is to be ended.

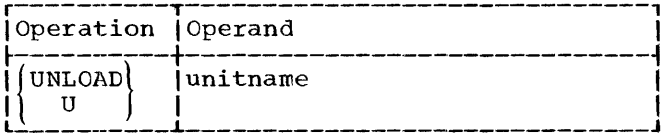
DSNAME
specifies that the system is to stop the display of the names of non-temporary data sets as initiated by the DSNAME parameter of the DISPLAY command. For more information, see the discussion of the DISPLAY command.

UNLOAD -- Prepare Volume for Demounting

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 demounting.

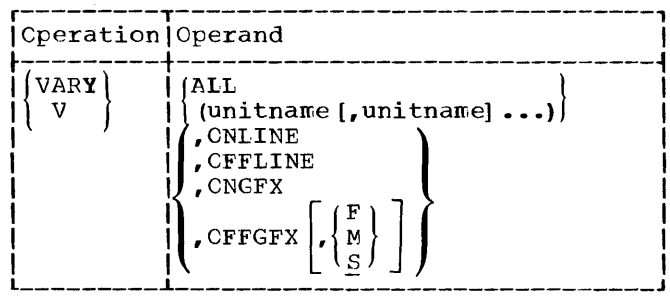
When the volume is ready to be demounted, you will receive a message. The message may not be received until the current job is completed.



unitname
specifies the unit address of the input/output device to be prepared for demounting. When using 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 unit name 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 input/output devices (other than a communication line) into an online or offline status. This command is also used to make the computer's 2250 display units available or unavailable for graphic job control operations.



ALL
specifies that all 2250s that were identified for GJP operations during system generation are to be made available (ONGFX) or unavailable (OFFGFX) for graphic job control operations.

unitname
specifies the unit address of either the input/output devices whose status are to be changed or the 2250s that are to be made available or unavailable for graphic job control operations.

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 address -- 263/8, for example, where the 8 is the number of the particular bin being addressed.

CNLINE
specifies that the input/output devices identified in this command are to be made available for allocation by the job scheduler to problem programs.

OFFLINE
specifies that the input/output devices identified in this command are to be removed from the recognition of the job scheduler, and that any further allocation of the devices to problem programs and system tasks is to be prevented. If the devices are in use (allocated to a problem program or to an input reader or output writer), their status is not changed until all the current users have finished with the devices. 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 or, if an appropriate system message is received, by issuing REPLY id,'unitname'.

ONGFX
specifies that the 2250s identified in this command are to be made available for graphic job control operations.

OFFGFX
specifies that the 2250s identified in this command are to be made unavailable for graphic job control operations.

F
(used with OFFGFX) designates that jobs being processed on the specified 2250s are to be ended immediately without notifying the 2250 users that this ending has occurred (F = fast stop). The system blanks the screens of the specified 2250s. A printed record of previous job control operations cannot be obtained.

M
(used with OFFGFX) designates that the jobs being processed on the specified 2250s are to be ended immediately and the user is to be requested to log off (M = medium stop). A printed record of previous control operations can be obtained.

S
(used with OFFGFX) designates that jobs currently being defined for or processed on the 2250s identified in this command are to be allowed to reach normal or abnormal end before the 2250 users are requested to log off (S = slow stop). A printed record of previous job control operations can be obtained. The operand is underscored to indicate that the system assumes S if none of the parameters F, M, and S has been specified.

Examples of VARY commands for IBM 2250 display units:

```
VARY ALL,ONGFX
```

This example causes all 2250s that were identified during system generation to be made available for graphic job control operations.

```
VARY (1E0,106),OFFGFX,S
```

This example causes the 2250 display units 1E0 and 106, which were made available by the command in the preceding example, to be made unavailable for graphic job control operations when the current jobs have been completed.

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 5, which describes general techniques.

HOW TO START A SYSTEM WITH MFT

1. Set LOAD UNIT to the channel, control unit, and device to address the system residence volume.
2. Hit LOAD 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 hit LOAD again.
3. Specify System Parameters. 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 5.
4. Wait for a CHANGE PARTITIONS? message and the WAIT light to go on. If the WAIT light goes on without a CHANGE PARTITIONS? message, display the current PSW, and follow the directions for the error code you will find in the low-order 12 bits 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.
5. After CHANGE PARTITIONS? is issued, you may change the partition definitions which were established at system

generation. See Redefining Partitions earlier in this chapter.

6. If partition redefinition is not desired, or after it has been terminated by END, the system issues the READY message followed by a display of the automatic START commands. Enter the SET command using the AUTO=operand to specify which automatic START commands you wish to accept. See the SET command for a description of the AUTO=parameter.

7. If you specified the formatting parameter [(Q=(unitname,[F]))] in the SET command the system now issues SPECIFY JOB QUEUE PARAMETERS. For a description of these parameters and how to alter them see "How to Specify Job Queue Parameters" in Chapter 5.

8. Enter any required START commands which have not been started automatically, and enter the DISPLAY command with the JOBNAMES parameter.

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, don't 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 job placed on the hold queue earlier.

HOW TO DETERMINE SYSTEM STATUS

Information describing current system activity is necessary to detect potential problems, and whenever possible, to prevent them from occurring. The commands listed below are particularly useful.

DISPLAY A Command

The DISPLAY A (active) command is your primary means of finding out what the system is doing. Use it frequently. Entering DISPLAY A causes each job name and associated stepname and the active system task and the partition in which it is running to be displayed on the console.

If one job seems to be running longer than it should (the job name continually appears on the console when the DISPLAY A is entered), this job may be contending for some resource such as:

- a device (check all operator messages; you may have missed a mount request) or
- a data set (the initiator is waiting for a data set).

Regularly check the console sheet for messages that may need replies (the DISPLAY R command will help you determine if there are unanswered messages). It may be necessary to CANCEL a job if an interlock condition exists; i.e., one job may be waiting for a data set that has been allocated to another job, and, in turn, the first job has a data set that the second job is waiting for. Neither job can continue until one of them relinquishes its data set. This is known as a system interlock. You should CANCEL one of the jobs and reenter it into the input stream.

DISPLAY JOBNAMES Command

This command gives you continuing data on jobs starting and stopping. After DISPLAY JOBNAMES is entered, message IEF403I 'jobname STARTED' is issued during initiation of each job. During termination, message IEF404I 'jobname ENDED' is issued to

notify you that the job has ended. Allocation of unit record equipment is also displayed when each job step starts.

DISPLAY N Command

Jobs assigned an unserviced CLASS parameter on the JOB card remain on their input queue indefinitely. Therefore, you should enter DISPLAY N periodically to obtain a listing of the jobs on the hold, input, and output work queues.

If there are jobs on input work queues that have no partitions assigned to service them, a partition can be redefined to service these classes (see "Redefining Partitions.") In addition, if you want to find out whether a job previously entered in the input stream has been run, enter DISPIAY N. If the job is not on any of the queues, it is either being run, or has already been run.

DISPLAY Q Command

If you have not started a resident writer (by entering a START command for a partition that has a job class identifier of WTR) DISPLAY Q should be entered periodically to determine the status of the output queues. If the number of entries in the output queues increases steadily, you should try to clean out the queues. This can be done by starting one or more writers in problem program partitions, specifying the output classes that have the most work on their queues.

When the number of entries on the queues has been reduced (enter DISPLAY Q again), you can STOP the writer, and the partition will again be available to process user jobs.

If you have started a resident reader (by entering a START command for a partition that has a job class identifier of RDR), DISPLAY Q should also be entered periodically to determine the status of the input queues. If the number of entries in the input queues increases steadily, you can redefine the resident reader partition to a problem program partition (with valid job class identifiers, A-O) that will process the user jobs. You can also issue a STOP command for any readers that you started in problem program partitions. This makes the partition available to process user jobs.

When the number of entries on the input queues has decreased, you can restart the reader in a problem program partition, or redefine the problem program partition back to a resident reader (see "Redefining Partitions" earlier in this chapter).

Low System Activity

Low system activity (indicated by the wait light frequently being on) can be due to the following conditions:

- Lack of system direct access space. Readers that do not read to end-of-file may have been delayed waiting for direct access space for the jobs and their associated data. Also, jobs that are waiting for allocation frequently cannot be assigned system output space. Both conditions can occur in normal system operation. They persist until space is made available as a result of writing the output data sets.

  If you have started several readers and writers in the system, the above delays may be reduced by entering STCP commands for readers to make direct access space available for output data sets. STCP commands can also be entered for writers which make direct access space available for initiation of new jobs.

- Mount Requests. When messages are written on the console to mount certain volumes, mount these volumes immediately. If you cannot do this, because of a lack of tape drives or disk drives, cancel the job.

- Small Partition Interlocks. If the system has only one scheduler-size partition and the system enters an enabled wait state, it may be in a small partition interlock; i.e., the initiator may be waiting for a data set that has been allocated to a job in a small partition. You will receive message IEF244I 'UNABLE TO ALLOCATE FROM AVAILABLE DEVICES'. You should cancel the job in the small partition, and reenter it in the input stream.

  In a Shared DASD system (see Chapter 5) where two or more systems are sharing the same device, an interlock can be caused by a programming error. This situation is usually characterized by all CPUs being in an enabled wait state. In this case, it may be necessary to cancel a job in only one of the systems to resolve the interlock. If not all CPUs are in an enabled wait state, proceed with caution; a system could simply be waiting for a device to be released.

HOW TO CONTROL JOBS THROUGH HOLD AND RELEASE COMMANDS

In MFT, in addition to the control provided by the job class facility, you can use HCLD and RELEASE commands to control

scheduling of jobs. All jobs of one type (e.g., production) initially can be put on the HOLD queue in two ways:

1. Use the TYPRUN=HOLD parameter on the JOB cards.

2. After you START a reader enter a HOLD command for the jobs that you want to be put on the HOLD queue. Do this before you enter any START INIT command.

You can then RELEASE these jobs one at a time (e.g., when the first production job is terminated, the next can be released). DISPLAY JOBNAMES can be extremely helpful in determining when jobs end, even though production-type jobs usually end in a predictable manner (rewinding tapes, receiving KEEP messages, etc.).

The HOLD command remains in effect until a RELEASE command is entered, even if the system is restarted. Jobs that were held individually or as part of the queues are still held after the system is restarted. They must be made available by entering the appropriate RELEASE command.

HOW TO SPECIFY THE USE OF THE GRAPHIC JOB PROCESSOR

The graphic job processor (GJP) is a program that displays user job control information at an IBM 2250 Display Unit. The user responds to the displays by entering requested information, by selecting appropriate options, or both. GJP converts the information entered by the 2250 user into job control statements and passes them to the operating system to initiate the job.

System Considerations When Using GJP

A separate partition must be made available for the graphics interface task (GXF) and for each 2250 that is to be used with GJP. The GFX partition must contain at least 10K bytes. Each 2250 partition must contain at least 60K bytes and should be time-sliced. Partitions used for GJPs that are time-sliced must be contiguous.

After GJP has been initiated, no other jobs should use the job classes, SYSOUT classes, or 2250s that have been assigned for GJP use.

After GJP has been stopped, the partitions to which it has been assigned are available for other uses and the special consideration described in the preceding paragraphs are no longer required.

Starting GJP

The first step in using GJP is to bring GFX into the operating system and to activate GFX. To do this, you must enter a START GFX command of a format as follows:

$$\left\{\begin{matrix}\text{START}\\ \text{S}\end{matrix}\right\}\text{GFX.Pn,,,(gfx=option,...,gfx=option)}$$

The gfx=option parameters are described under "GFX Options."

After GFX has been activated, you must enter a VARY ONGFX command to designate the 2250s that are to be made available for GJP.

Stopping GJP

The VARY OFFGFX command can be used at any time to designate 2250s that are no longer to be used with GJP. If you include the S (slow stop) parameter in your command (explicitly or by default), any jobs currently being defined or processed on the 2250s are allowed to reach their normal or abnormal completion.

If you specify the M (medium stop) parameter or F (fast stop) parameter in your command, you force an immediate ending of GJP activity at the specified 2250s. (See the description of the VARY command for the effects of these parameters.) If necessary, you can issue a VARY OFFGFX command containing an M or F parameter for a 2250 after a VARY OFFGFX command containing an S parameter has been issued for that unit.

A STOP GFX command can be used at any time to remove GFX from main storage. A STOP GFX command does not end GFX until GJP activity at all 2250s has ended. If GJP activity continues and you want the STOP command to take effect quickly, you can issue appropriate VARY OFFGFX commands. VARY OFFGFX commands can be issued even though a STOP GFX command has already been entered.

After a STOP GFX command has been issued and all GJP activity has ended, GFX is ended and the partitions and 2250s allocated for the ended GJP are freed.

GFX Options

These are special keyword=option parameters of the format gfx=option. They are used in the START GFX command to override options selected during system generation and can be specified in any sequence. Options selected during system generation are used for any parameters not specified.

The following gfx=option parameters can be used:

PRT=printer output class
PCH=punch output class
MSGF=foreground message class
MSGB=background message class
PRIF=foregound priority
PRIB=background priority
CLSF=foreground job class
CLSB=background job class

PRT=printer output class
specifies the class name that identifies the SYSOUT class for printed output from jobs defined at a 2250 by means of GJP.

PCH=punch output class
specifies the class name that identifies the SYSOUT class for punched output from jobs defined at a 2250 by means of GJP.

MSGF=foreground message class
specifies the class name that identifies the SYSOUT class for messages pertaining to foreground jobs defined at a 2250 by means of GJP. This class name should be unique to the system while GFX is active.

MSGB=background message class
specifies the class name that identifies the SYSOUT class for messages pertaining to background jobs at a 2250 by means of GJP.

PRIF=foreground priority
specifies the priority to be used for foreground jobs defined at a 2250 by means of GJP. The priority is a 1-digit or 2-digit integer (0-13) that determines the scheduling priority for these jobs.

PRIB=background priority
specifies the priority to be used for background jobs defined at a 2250 by means of GJP. The priority is a 1-digit or 2-digit integer (0-13) that determines the scheduling priority for these jobs.

CLSF=foreground job class
specifies the job class (A through C) for the graphic job processors and for foreground jobs defined at the 2250s by means of GJP. The job class determines the partitions in which the graphic job processors and foreground jobs are executed and thereby establishes the dispatching priority for these jobs.

CLSB=background job class
specifies the job class (A through C) for background jobs defined at the 2250s by means of GJP. This job class must include at least one partition for each 2250 that will use GJP. This job class determines the partitions in which background jobs are executed and thereby establishes the dispatching priorities for background jobs.

HOW TO PRINT A SYSOUT TAPE

In systems with MFT, use the utility program called IEBGENER to print a SYSCUT 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   DSNAME=SYSOUT,UNIT=2400,   c
//                DISP=(OLD,DELETE),         c
//                VOLUME=SER=(VOL1,...,VCLn)
//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 SYSCUT writer, and must be specified in the same order as they were created.

The job scheduler in a system with MVT (Figure 5) is similar to the job scheduler in systems with the primary control program. 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. You give commands to the system, and receive messages from it, through the console. The most commonly used console I/O device is the IBM 1052 Printer-Keyboard. However, other devices can be used including an IBM 2250 Display Unit Model 1 with the IBM System/360 Models 50, 65, and 75.

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. With an IBM System/360 Model 50, 65, or 75, either the primary console or the alternate console (but not both) may be an IBM 2250 Display Unit Model 1.

The primary console is active until you shift to the alternate by hitting 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 5.

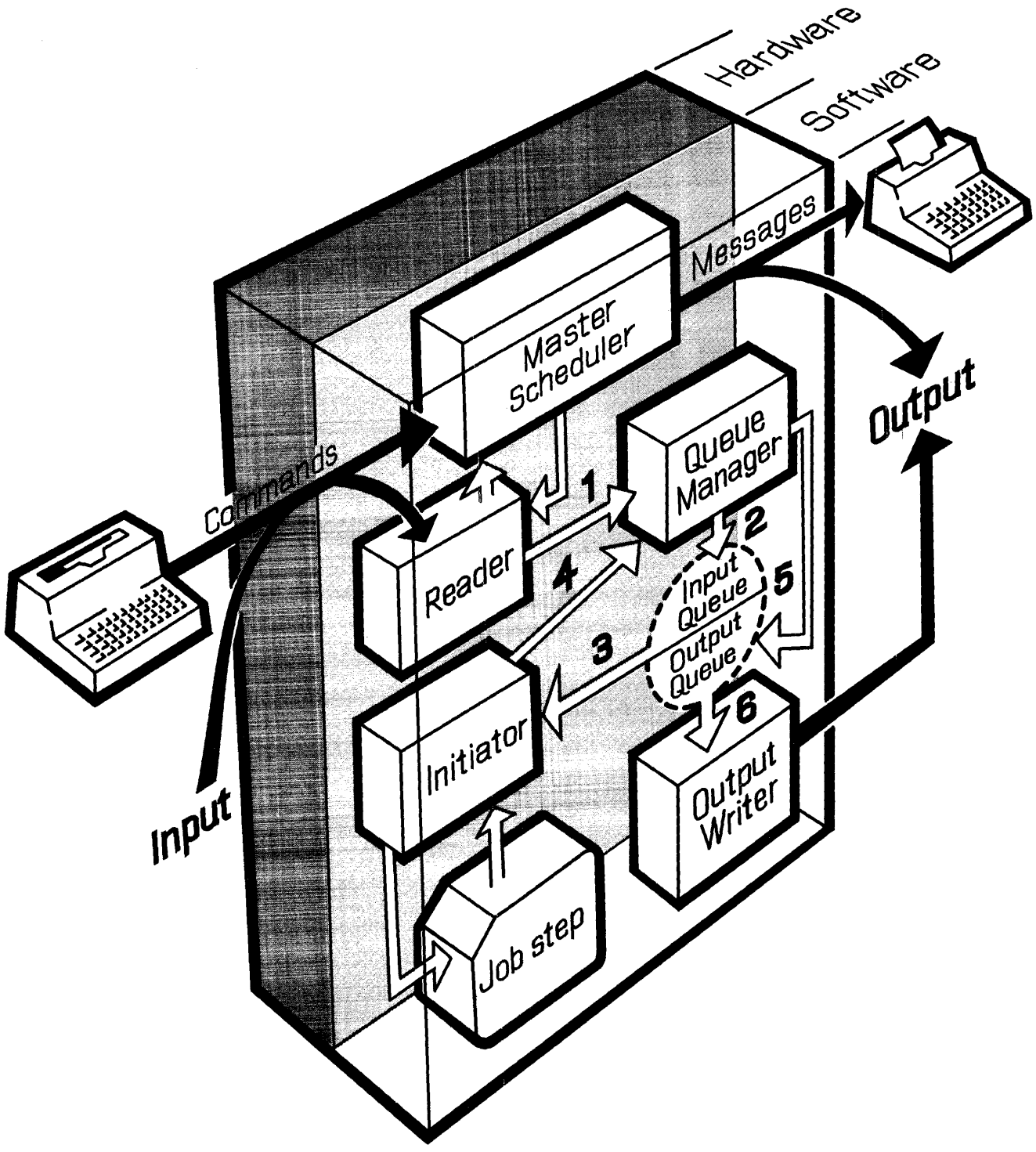


Figure 5. How the Scheduler Works in Systems With MVT

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, hit 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 holding down the alternate coding key and hitting the numeric 5 key.

You can cancel a typed line by holding down the alternate coding key and hitting the numeric 0 key.

To give the system a command through an IBM 2250 Display Unit, type in your command on the keyboard. The number of typed characters per command cannot exceed 126. To indicate the end of a command, signal END by holding down the alternate coding key and hitting the numeric 5 key. You can cancel a typed line by holding down the alternate coding key and hitting 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) hit the card reader's STOP key, place the command in the reader, and hold down the START key and hit the EOF key.

You can also enter most commands through a SYSIN (system input) stream by using command statements.

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 aren't 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 hit the LOAD key on the panel.

Hitting 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 5.

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 12 bits 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 and Codes.

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 5, under the heading "How to Specify System Parameters."

If no changes are to be made, issue REPLY,id,'U', or simply signal EOB.

After NIP completes its preparation of the system, it passes control to the master scheduler. You'll 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 low-order 12 bits 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" in Chapter 5.

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.identifier
START RDR.identifier
START INIT.identifier
```

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.identifier
START INIT.identifier
START INIT.identifier
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 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, a STOP GFX for the graphic interface task (if such a task was initiated), 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'll 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's 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.

Job control information is placed in the input queue for use by the job scheduler in selecting the jobs and job steps to be processed.

Job Classes

Jobs can be grouped, depending upon your installation needs, into as many as 15 different job classes (A-O).

Each class has a one-character classname and includes all job control information for input jobs associated with the classname. This permits grouping of jobs with a common characteristic. The characteristic might be a job with much input/output, a job requiring a large amount of CPU processing, or possibly one which requires special control volumes.

The CLASS for a job is specified by the CLASS parameter of the JOB statement.

Initiators

An initiator is that part of the job scheduler which selects jobs and job steps to be processed.

By using START INIT, or any other procedure name which your system programmer may supply, you specify the classnames that the initiator is to process.

For example, to process jobs from the input queue which have been grouped in class D, you would issue:

```
START INIT,,,D
```

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 one-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.

If your system has a universal character set (UCS) printer that will be used as an output writer, you must assign a separate output class to each character set image stored in the system library. This assignment preserves the identity of character set-dependent data through the SYSOUT stream. The output is grouped by character set-code which minimizes your changing of printer chains and trains.

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 printed output, such as a summary dump of disk records for auditing.

The SYSCUT 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 SYSCUT data set is in the class specified by the SYSCUT 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 SYSCUT 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 class names and devices, by naming (a) the device to be allocated to the output writer and (b) the class names 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 6).

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.

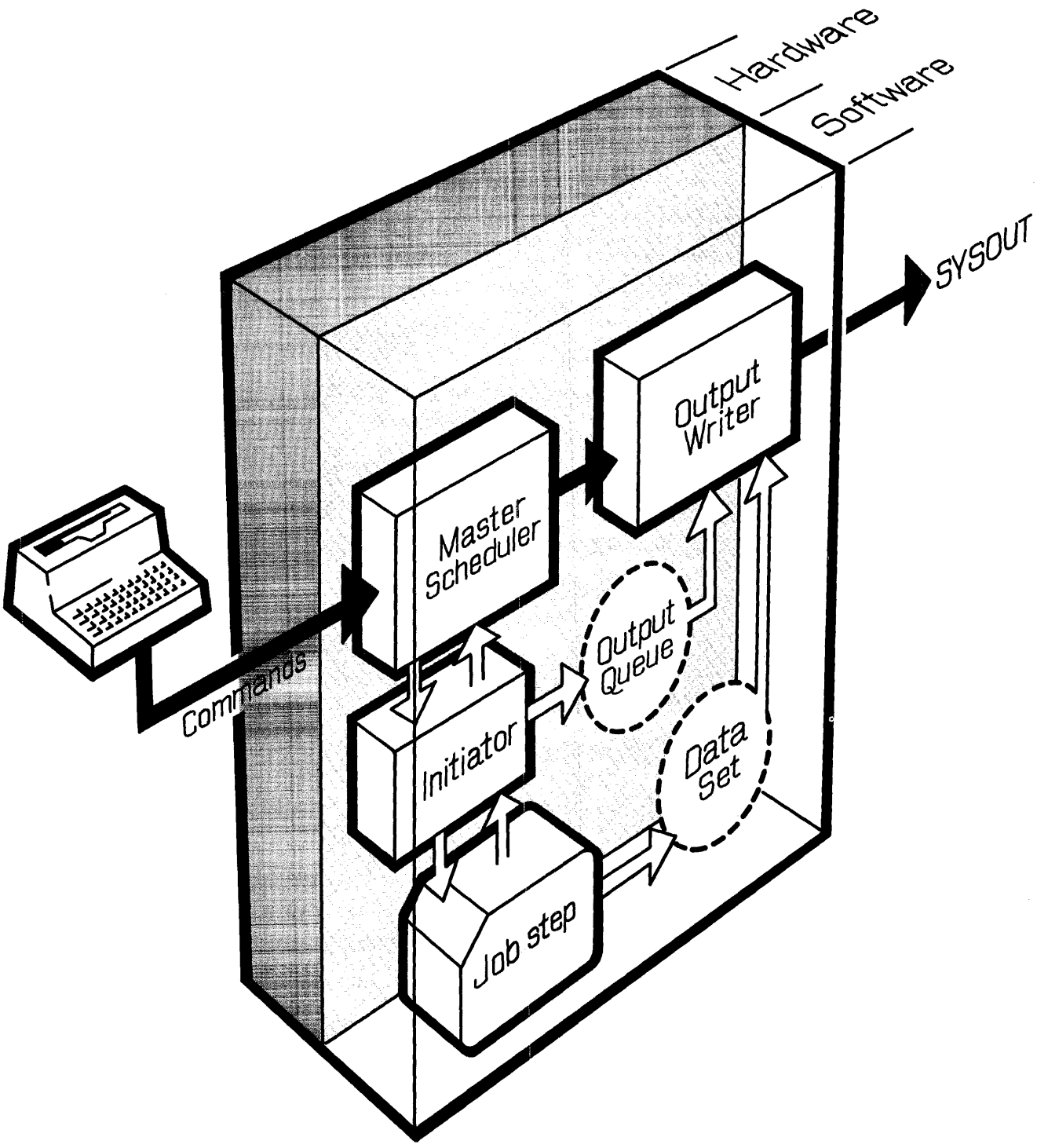


Figure 6. Output Process in Systems with MVT

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 don't 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.

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 demount 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.

Never mount a blank tape volume, unless specifically directed to do so.

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 standard labels on 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 demounted, the system unloads the devices, if possible, and writes you messages identifying the volumes being unloaded.

When you receive a mounting message for a 2321 data cell, hit 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 don't have to hit RESET -- closing the door performs the same function in this case.

Occasionally you may be asked by your installation not to mount volumes on certain devices, and not to make those devices ready, because you are running a version of the operating system that was generated for a slightly different set of devices.

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 demounted 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 alphameric 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.

Shared DASD Option: If your installation is using a system with this option, be sure to read the section "How to Use the Shared DASD Option" in Chapter 5.

CPERATCR CCMMANDS

This section contains a description of the commands you use to control 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
HCLD	H	START	S
LOG	L	STOP	P
MODIFY	F	UNLOAD	U
MCUNT	M	VARY	V
RELEASE	A	WRITELCG	W

*This command cannot be entered into the input stream.

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. But 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 one of the input work queues 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 its associated 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 for system tasks. The STOP command must be issued at this time.

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.

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

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. There is no indicative dump facility in systems with MVT.

unitname
specifies the name of an I/O device. The system will stop the output writer from completing the writing of the data set being written on the named device. The output writer automatically continues processing further output.

identifier
specifies the identifier of a system task to be terminated during allocation. The identifier may be one of the following: The identifier specified in the START command; if no identifier was specified, the unit requested in the START command; if neither unit nor identifier is specified in the START command, the unit requested in the procedure associated with this system task; or, if none of the above is specified, the procname in the START command.

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.)

This 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.

In addition, the DISPLAY command is used to cause a console display of the time of day and the date, of job status or of the activity of tasks within the system, and of the names of non-temporary data sets in mount and K (keep) type demount messages.

Operation	Operand
{DISPLAY / D}	jobname JOBNAMES [,T] T STATUS A R Q [=list] N [=list] DSNAME

jobname
: specifies the name of the job for which the following is to be displayed: job name; class; job priority; type of the queue the job is in - JOB Q, HOLD Q, or SOUTQ (SYSOUT queue); and position in the queue. The maximum length of a job name is eight characters. If JOBNAMES, STATUS, T, A, R, Q, N, or DSNAME is used as a job name, it must be in parentheses.

JOBNAMES [,T]
: 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.

  If the T parameter is used in conjunction with the JOBNAMES parameter, the system displays the time of the day in addition to the job names. The time is shown in the format:

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

T
: specifies that the time of day and the date are to be displayed in the following format:

```
hh.mm.ss                yy.ddd
|  |  |                  |   |
|  |  └--Seconds         |   |
|  └-----Minutes         |   └--Day
└--------Hours (00-23)   └------Year
```

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 end and job end.

A
: specifies that the system is to display information about all the jobs and job steps recognized by the supervisor as tasks -- that is, those jobs and job steps that have been processed by the initiator and have had a task control block (TCB) made for them by the supervisor.

  The names of the job and job step associated with each task are displayed, as well as a number telling how many subordinate tasks (those pieces of work that support the main or job step tasks) are operating within the same region of main storage, the beginning and end addresses of that region, and the amount of supervisor queue space used for system control blocks related to the main task.

  If rollout is included in the system, the display will indicate whether the region is borrowed or rolled out. In addition, similar information about related system tasks, such as tasks associated with readers and output writers, will be displayed.

R
: specifies that the system is to display

  - The id of each message that required a reply and has not yet been replied to,
  - The unit number of each device for which a mount message has been issued but has not been complied with, and
  - An indication whether or not any AVR mount messages are pending.

  When you use the DISPLAY R command, the system displays the requested information as message IEE110A if any action on your part (reply to messages or mount volumes) is yet required or as message IEE111I if you have complied with all system requests.

Q
specifies that a listing of the number of entries on each of the non-empty input, hold, and output queues is to be displayed.

N
specifies that a listing of job names on the hold, input, and output queues is to be displayed.

list
specifies any combination of up to four of the following items:

specific input work queue name (job class A through O)

SOUT (system output queues collectively)

HOLD (system hold queue)

If list includes more than one item, you must separate the specified items by commas and enclose them in parentheses. If no list value is specified, all 15 input work queues, the hold queue, and the output queue are assumed.

DSNAME
specifies that the system is to display, within the mount and K (keep) type demount messages, the name of the first non-temporary data set allocated to the volume to which the messages refer.

Note: When you use DISPLAY A, the system response will be a message, IEE102I, listing the information in the following order: j s t x a1 a2 qs y.

j will be one of the following: the name of a job attached by an initiator; MASTER SCHEDULER for the master task; the procname for a system task such as WTR or RDR; the initiator identifier for an initiator; or STARTING for a system task being allocated.

s will be one of the following: the name of the job step; the procedure step name if the t field is not blank; blanks if j is MASTER SCHEDULER or STARTING; an identifier, if j is the procname for a system task; a job name or blanks, if j is an initiator identifier (blank if the initiator is waiting for work).

t will be either the name of the step that called the cataloged procedure whose name is shown in the s field or blanks if the step is not a cataloged procedure.

x will be the number, in decimal, of subtasks operating in the region.

a1 will be either the beginning address of the region, or zeros if the initiator is waiting for work, main storage or a data set.

a2 will be either the end address of the region, or zeros if the initiator is waiting for work, main storage or a data set.

qs will be the number of bytes, in decimal, of supervisor queue space required.

y will be an indication that the region is either owned, borrowed, or rolled out.

blank = owned
B = borrowed
R = rolled out

If j is an initiator identifier, s is a job name, and region limits a1 and a2 are given, the initiator is starting or ending a step.

If j is an initiator identifier, s is a jobname, but a1 and a2 are zeros, the initiator is waiting for main storage or a data set.

If j is an initiator identifier, s is blank, and region limits are zero, the initiator is waiting for work; if jobs are being held, you can then 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. This command cannot be entered in the input stream.

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 WRITELOG CLOSE command.

Operation	Operand
HALT Z	EOD

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 one of the input work queues, 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 specified input work queues 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 that is in a queue which is held because of a HOLD Q command.

Operation	Operand
{HOLD / H}	{jobname / Q[=list]}

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 specified or implied input work queue is to be suspended.

list
: specifies any combination of up to four input work queue names (job class A through O). If no list value is specified, all 15 input work queues are assumed. If list includes more than one item, you must separate the specified items by commas and enclose them in parentheses.

LOG -- Store Information in Log

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

Operation	Operand
{LOG / L}	'text'

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 System Characteristics

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

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}	[procname.]identifier [,CLASS=classnames] [,PAUSE={FORMS / DATASET}]

procname
: specifies the name of the procedure started with a START command.

identifier
: specifies the identification of the system task as defined by the START command.

CLASS=classnames
: specifies one to eight single-character names of the classes to be associated with the initiator or output writer, for example, CLASS=ABCD. If more than one class name is specified, the classes are treated on a priority basis, where the left-most character indicates the class to be processed first. Note: The initiator and output writer will also

accept class names as a series of characters in parentheses and separated by commas; for example, CLASS=(A,B,C,D). But the form ABCD is preferred to (A,B,C,D).

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 demountings and remountings of that volume.

Operation	Operand
{MOUNT / M}	{unitname / devicename} {,VOL=(NL,serial) / ,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. 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 unit name 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.

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 alphameric 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 alphameric serial number, up to six characters long, is used in label checking and for allocation references.

USE=STCRAGE 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.

RELEASE -- Make Job Available for Selection

The RELEASE command is used to resume job selection that has been suspended by the HCLD 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 or TYPRUN=HOLD on the JOB card, issue a RELEASE jobname command.

To release jobs held because they were on the specified 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 that is in a queue which is held because of a HOLD Q command.

Operation	Operand
{RELEASE / A}	{jobname / Q [=list]}

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.

list
specifies any combination of up to four input work queue names (job class A through O). If no list value is specified, all 15 input work queues are assumed. If list includes more than one item, you must separate them by commas and enclose them in parentheses.

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 6.

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 number 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 don't 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:

```
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) or (unitname,F) or (,F)
specifies (1) the name of the direct access device (other than an IBM 2321) on which the volume containing the input work queue (SYS1.SYSJCBQE) resides or (2) that the system is to format the input work queue, or both.

This parameter is used only in the first SET command after IPL.

Space for the input work queue must have already been allocated on the volume which is mounted on the specified device.

You need not specify a 3-character unit name if one of the following conditions exists:

- The SYS1.SYSJOBQE data set is cataloged,
- the volume containing the SYS1.SYSJOBQE data set resides on the device that has been specified at the time of system generation.
- the SYS1.SYSJOBQE data set is contained on the system residence volume.

If the system is to format the input work queue prior to the first job initiation, you must specify ,F following or without the 3-character unit name. For example, Q=(unitname,F) specifies that the system is to format the input work queue on the volume residing on the direct access device referred to by unitname; Q=(,F) specifies that the system is to format the input work queue either on the volume to which the SYS1.SYSJOBQE data set is cataloged, or on the volume that resides on the device specified at the time of system generation, or on the system residence volume.

If this parameter is not given, the system searches the system catalog (SYSCTLG) to determine and select the volume that contains the SYS1.SYSJOBQE data set. If this data set is not cataloged, the system uses the values provided at the time of system generation, if any such values were provided. If no such values were provided, the system assumes that the data set is contained on the system residence volume.

The system issues an error message when either of the following conditions exists:

1. The volume to which the SYS1.SYSJOBQE data set is cataloged is not mounted.
2. The system cannot locate the SYS1.SYSJOBQE data set on the selected volume.

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 searches the system catalog (SYSCTLG) to determine and select the volume that contains the SYS1.PRCCLIB data set. If this data set is not cataloged, the system uses the values provided at the time of system generation, if any such values were provided. If no such values were provided, the system assumes that the data set is contained on the system residence volume.

The system issues an error message if either of the following conditions exists:

1. The volume to which the SYS1.PROCLIB data set is cataloged is not mounted.
2. The system cannot locate the SYS1.PROCLIB data set on the selected 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=NCNE.

START -- Start System Process

The START command is used to start initiators, readers, writers, and the graphics interface task.

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

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, WTR for an output writer INIT for an initiator, and GFX for the graphic interface task. For additional information on the graphic job processor, see the discussion under "How to Specify the Use of the Graphic Job Processor."

identifier
specifies an optional name of up to eight characters, the first of which must be alphabetical. You can add a name to identify a particular procname for use with MODIFY and STOP commands when you have started more than one procname with the same name. You must not use the same identifier with different initiator procnames.

The identifier to be used in the MODIFY and STOP commands is derived in the following way. When an identifier is specified in the START command, it unconditionally becomes the identifier. If an identifier is not specified, the identifier is assigned either the unitname of the device allocated to the system task started by the START command or, when no unit is allocated, the procname specified in the START command. The following table indicates the identifier associated with each started system task.

START Command	Unit allocated	identifier
S RDR.A,00C	00C	A
S RDR,00C	00C	00C
S INIT	none	INIT
S WTR	00E (from proc)	00E

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.

If you don't use this parameter but specify one or both of the parameters volumeserial and parmvalue, you must indicate the absense of the device parameter by a comma. (See the example under "parmvalue" below.)

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.

If you don't use this parameter but specify a parmvalue parameter, you must indicate the absence of the volumeserial parameter by a comma. (See the example under "parmvalue" below.)

parmvalue
specifies parameter values to be passed to the program receiving control as a result of the START command. If parmvalue contains any nonalphameric 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 the class name parmvalue used by an initiator or output writer, for example, S WTR.CNE,00E,,ABC.

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 found.

A class name parmvalue may be used when starting an initiator or output writer: use this parmvalue to limit the classes the initiator or writer is to process. The class name parmvalue may be from one to eight characters which represent the classes to be processed. The specified classes are treated on a priority basis where the left-most character indicates the queue to be processed first. A class name parmvalue entry causes all class names specified in the cataloged procedure to be overridden.

The initiator and output writer will also accept a class name as a series of characters in parentheses and separated by commas; for example, S WTR.ONE,00E,,(A,B,C). But the form ABC is preferred to (A,B,C).

If you specify this parameter but none or only one of the parameters devicename and volumeserial, you must indicate the absence of the latter two by commas. Example: S RDR,2400,,PAYROLL.

gfx=option
specifies a new value for and overrides the corresponding GFX option value selected at the time of system generation. You can override just one or more or all of the GFX option values specified at the time of system generation.. For a description of the gfx=option parameters, see "How to Specify the Use of the Graphic Job Processor."

In a START command for a graphic interface task, the gfx=option takes the position of the parmvalue parameter. Therefore whenever you issue a START command specifying a GFX option value, you must indicate by commas the absence of the parameters devicename and volumeserial. Example: S GFX,,,(gfx=option, gfx=option).

You can omit the enclosing parentheses if you specify only one GFX option value.

You cannot specify a keyword=option parameter in a START command for a graphic interface task.

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 DSNAME=name 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=volumeid. 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.

STOP -- Stop System Process

The STOP command is used to stop the operation of an input reader, an output writer, an initiator, or the graphics interface task, or to stop a console display of job and data set names.

When you use this command to stop a function other than a display, the procedure does not stop immediately. Instead the system begins the stopping after the procedure finishes handling its current task. After a reader is started, it must process one job before a STOP RDR command will take effect. In addition, input readers stop automatically on meeting an end-of-file condition.

Operation	Operand
{STOP / P}	{[procname.]identifier / JCBNAMES / STATUS / DSNAME}

procname
specifies the name of the procedure started with START command.

identifier
specifies the identification of the system task as defined by the START command.

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

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

DSNAME
specifies that the system is to stop the display of the names of nontemporary data sets as initiated by the DSNAME parameter of the DISPLAY command. For more information, see the discussion of the DISPLAY command.

Note: The STOP command will accept the form procname,identifier although procname.identifier is the preferred form.

UNLOAD -- Prepare Volume for Demounting

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 demounting. When the volume is ready to be demounted, you will receive a message.

(The message may not be received until the current job is completed.)

Operation	Operand
{UNLOAD / U}	unitname

unitname
specifies the unit address of the input/output device to be prepared for demounting. 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 unit name 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 input/output devices (other than a communication line) into an online or offline status. This command is also used to make the computer's 2250 display units available or unavailable for graphic job control operations.

Operation	Operand
{VARY / V}	{ALL / (unitname [,unitname] ...)} {,ONLINE / ,OFFLINE / ,ONGFX / ,OFFGFX [,{F / M / S}]}

ALL
specifies that all 2250s that were identified for GJP operations during system generation are to be made available (ONGFX) or unavailable (OFFGFX) for graphic job control operations.

unitname
specifies the unit address of either the input/output devices whose status are to be changed or the 2250s that are to be made available or unavailable for graphic job control operations.

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.

CNLINE
specifies that the input/output devices identified in this command are to be made available for allocation by the job scheduler to problem programs.

CFFLINE
specifies that the input/output devices identified in this command are to be removed from the recognition of the job scheduler, and that any further allocation of the devices to problem programs and system tasks is to be prevented. If the devices are 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 devices. 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 or, if an appropriate system message is received, by issuing REPLY id,'unitname'.

CNGFX
specifies that the 2250s identified in this command are to be made available for graphic job control operations.

CFFGFX
specifies that the 2250s identified in this command are to be made unavailable for graphic job control operations.

F
(used with OFFGFX) designates that jobs being processed on the specified 2250s are to be ended immediately without notifying the 2250 users that this ending has occurred (F = fast stop). The system blanks the screens of the specified 2250s. A printed record of previous job control operations cannot be obtained.

M
(used with OFFGFX) designates that the jobs being processed on the specified 2250s are to be ended immediately and the user is to be requested to log off (M = medium stop). A printed record of previous control operations can be obtained.

S
(used with OFFGFX) designates that jobs currently being defined for or processed on the 2250s identified in

this command are to be allowed to reach normal or abnormal end before the 2250 users are requested to log off (S = slow stop). A printed record of previous job control operations can be obtained. The operand is underscored to indicate that the system assumes S if none of the parameters F, M, and S has been specified.

Examples of VARY commands for IBM 2250 display units:

```
VARY ALL,ONGFX
```

This example causes all 2250s that were identified during system generation to be made available for graphic job control operations.

```
VARY (1E0,106),OFFGFX,S
```

This example causes the 2250 display units 1E0 and 106, which were made available by the command in the preceding example, to be made unavailable for graphic job control operations when the current jobs have been completed.

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 / W}	{classname / 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, skip to Chapter 5, which describes general techniques.

HOW TO START A SYSTEM WITH MVT

1. Set the LOAD UNIT switches to the channel, control unit, and device to address the system residence volume.

2. Hit 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 hit the LOAD key again.

3. Specify System Parameters. 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 Chapter 5.

4. 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 low-order 12 bits 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 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.

If you specified the formatting parameter Q=([unitname][,F]) in the SET command the system now issues SPECIFY JOB QUEUE PARAMETERS. For a description of these parameters and how to alter them see "How to Specify Job Queue Parameters" in Chapter 5.

Further commands can now be entered manually.

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 all operator 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 RDRs and WTRs	(to assure no main storage fragmentation)
CANCEL jobs	(cancel jobs in the reverse order of their importance; should be done only as a last resort)
DISPLAY ACTIVE	(should be used repeatedly to trace changing environment)
RELEASE Q	(to resume normal processing)

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

Regularly check the console sheet for messages that may need replies (the DISPLAY R command will help you determine if there are unanswered messages).

Reduced system activity can also be due to the following:

- Exhausted system direct access space. Readers that don't 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.
- In a Shared DASD system (see Chapter 5) where two or more systems are sharing the same device, an interlock has been caused by a programming error. This situation is usually characterized by all CPUs being in an enabled wait state. In this case, it may be necessary to cancel a job in only one of the systems to resolve the interlock.

  If not all CPUs are in an enabled wait state, proceed with caution; a system could simply be waiting for a device to be released.

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

In addition to the control provided by the job class facility, 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, don't 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 SPECIFY THE USE OF THE GRAPHIC JOB PROCESSOR

The graphic job processor (GJP) is a program that displays user job control information at an IBM 2250 Display Unit. The user responds to the displays by entering requested information, by selecting appropriate options, or both. GJP converts the information entered by the 2250 user into job control statements and passes them to the operating system to initiate the job.

System Considerations When Using GJP

To ensure uniform operation, the priority for GJP should be high and should be equivalent to a time-slice priority. Furthermore, the priority and SYSCUT classes assigned to GJP should not be assigned to any other job.

Starting GJP

The first step in using GJP is to bring the graphics interface task (GFX) into the operating system and activate GFX. To do so, you must enter a START GFX command of a format as follows:

```
{START}GFX[,,,(gfx=option,...,gfx=option)]
{  S  }
```

The gfx=option parameters are described under "GFX Options."

After GFX has been activated, you must enter a VARY CNGFX command to designate the 2250s that are to be made available for GJP.

Stopping GJP

The VARY OFFGFX command can be used at any time to designate 2250s that are no longer to be used with GJP. If you include the S (slow stop) parameter in your command (explicitly or by default), any jobs currently being defined or processed on the 2250s are allowed to reach their normal or abnormal completion.

If you specify the M (medium stop) or F (fast stop) parameter in your command, you

force an immediate ending of GJP activity at the specified 2250s. (See the description of the VARY command for the effects of these parameters.) If necessary, you can issue a VARY OFFGFX command containing an M or F parameter for a 2250 after a VARY OFFGFX command containing an S parameter has been issued for that unit.

A STOP GFX command can be used at any time to remove GFX from main storage. A STOP GFX command does not end GFX until GJP activity at all 2250s has ended. If GJP activity continues and you want the STOP command to take effect quickly, you can issue appropriate VARY OFFGFX commands. VARY OFFGFX commands can be issued even though a STOP GFX command has already been entered.

After a STOP GFX command has been issued and all GJP activity has ended, GFX is ended and the regions and 2250s allocated for the ended GJP are freed.

GFX Options

These are special keyword=option parameters of the format gfx=option. They are used in the START GFX command to override options selected during system generation and can be specified in any sequence. Options selected during system generation are used for any parameters not specified.

The following gfx=option parameters can be used:

PRT=printer output class
PCH=punch output class
MSGF=foreground message class
MSGB=background message class
PRIF=foreground priority
PRIB=background priority
CLSF=foreground job class
CLSB=background job class
RGNG=gjp region size
RGNF=foreground region size
RGNB=background region size

PRT=printer output class
specifies the class name that identifies the SYSOUT class for printed output from jobs defined at a 2250 by means of GJP.

PCH=punch output class
specifies the class name that identifies the SYSOUT class for punched output from jobs defined at a 2250 by means of GJP.

MSGF=foreground message class
specifies the class name that identifies the SYSOUT class for messages pertaining to foreground jobs defined at a 2250 by means of GJP. This class name should be unique to the system while GFX is active.

MSGB=background message class
specifies the class name that identifies the SYSOUT class for messages pertaining to background jobs defined at a 2250 by means of GJP.

PRIF=foreground priority
specifies the priority to be used for foreground jobs defined at a 2250 by means of GJP. The priority is a 1-digit or 2-digit integer (0-13) that determines the scheduling priority (within an input work queue) and the dispatching priority for these jobs.

PRIB=background priority
specifies the priority to be used for background jobs defined at a 2250 by means of GJP. The priority is a 1-digit or 2-digit integer (0-13) that determines the scheduling priority (within an input work queue) and the dispatching priority for these jobs.

CLSF=foreground job class
specifies the job class (A through C) for the graphic job processors and for foreground jobs defined at a 2250 by means of GJP. The job class determines the input work queue in which the jobs will be placed and thus determines (in combination with the priority) how quickly the jobs will be initiated.

CLSB=background job class
specifies the job class (A through C) for background jobs defined at a 2250 by means of GJP. The job class determines the input work queue in which background jobs will be placed and thus determines (in combination with priority) how quickly the jobs will be initiated.

RGNG=gjp region size
specifies the number of 1,024-byte (1K) blocks of main storage to be included in a region to be occupied by GJP. The "gjp region size" is specified with an integer. The region for GJP must be at least 60K bytes.

RGNF=foreground region size
specifies the number of 1,024-byte (1K) blocks of main storage to be included in regions in which foreground jobs, defined by means of GJP, are executed. The parameter is specified with an integer.

RGNB=background region size
specifies the number of 1,024-byte (1K) blocks of main storage to be included in regions in which background jobs, defined by means of GJP, are executed. The parameter is specified with an integer.

HOW TO PRINT A SYSOUT TAPE

In systems with MVT, use the utility program called IEBGENER to print 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   DSNAME=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	
2540-2		x

IBM 2250 USED AS AN OPERATOR'S CONSOLE

This section tells how to use the optional IBM 2250 Display Unit, Model 1, as a system operator's console on an IBM System/360 Model 50, 65, or 75.

With the IBM 2250, you can:

- Observe messages displayed by the system and the problem program.
- Reply to messages which require replies.
- Issue commands to the system.
- Request display of system status information.

In addition, a hard copy of messages, replies, and commands can be provided.

A 2250 that has been specified as your primary console can be switched from primary to alternate console status or vice versa. When the 2250 is the primary console, and either a 1052 Printer-Keyboard or a composite console consisting of a card reader and a printer is available as an alternate console, hard copy of commands and messages can be provided on the alternate console.

Using a 2250 as a primary console makes it unavailable for use on problem programs. You can make the 2250 available for problem program usage by placing it in the alternate console status. However, when a 2250 alternate console is allocated to a problem program, it cannot be used as a primary console until the program is ended.

CONSOLE DISPLAY

The basic display, illustrated in Figure 7, is divided into four general message display areas. The line numbers for each area are shown below in parentheses. These line numbers appear in the figure for illustration purposes only; they aren't in the actual screen display. The four general display areas are:

1. Write to Operator (WTO) (lines 3 through 39)
2. Write to Operator with Reply (WTOR) (lines 41 through 44)
3. Entry (lines 46 and 47)
4. Control (lines 49 and 50)

Write to Operator Message Display Area

The Write to Operator (WTO) message display area displays all messages issued by a WTO macro instruction in the control program or in a problem program.

The WTO area consists of 37 lines. Its header line is "*WTO MESSAGES" (see Figure 7, line 2). One 74-character line is allowed for each message. Messages containing more than 74 characters are cut short.

Each new message is entered at the bottom of the area. As each new message is entered, the old messages move up one line.

All messages in the WTO area must be deleted to make room for new messages. To delete a message, point the light sensor (commonly called a light "pen") at the message to be deleted, while at the same time pressing the foot pedal. This causes the message to be underlined and a set of action options to appear on the right-hand side of the line (see Figure 8, line 39). If necessary, the last characters of the message are temporarily overwritten. Then the light pen is used to select one of the following options:

YES -- delete the selected message. The message is erased and all messages displayed above it move down one line.

NC -- do not delete the selected message. The original message text is restored with no change.

ALL -- delete the selected message and all messages displayed above it. The selected messages are erased.

If the WTO message display area is filled, new messages are prevented from being displayed. These messages are not lost, but are held until older messages are deleted. An alarm is sounded and a warning message is displayed in the control display area (see Figure 9, line 49) to alert you that a new message cannot be displayed. The warning message is removed when the new message is displayed.

Also, the message hold option may be used to prevent new messages from being displayed. When this option is YES, all new messages will be held. (See "MSGE Hold Option" later in this chapter.)

Write to Operator With Reply Message Display Area

The Write to Operator with Reply (WTOR) message display area displays all messages issued by a WTOR macro instruction in the control program or in a problem program.

The WTOR area consists of four lines. (See Figure 7, lines 41 through 44). Its header line for is "*WTOR MESSAGES" (see Figure 7, line 40). One 74-character line is allowed for each message.

A new message is displayed on any available line in the WTOR area and on the bottom line of the WTO area if there is a line available in both areas. An alarm is sounded as each new message is displayed.

When your reply to a WTOR message is entered and accepted (see "Entry Message Display Area"), the corresponding message is deleted from the WTOR area and the reply is displayed in the WTO area.

You can delete WTOR messages by using the light pen in the same manner as WTO messages, except that just one message can be selected at one time and, therefore, only the YES and NO options appear on the right-hand side of the line. Messages are automatically deleted when the issuing task is ended.

A warning message, similar to the one shown for the WTO area, is displayed in the control display area and an alarm is sounded when a new message cannot be displayed. The warning message is deleted when the new message is displayed.

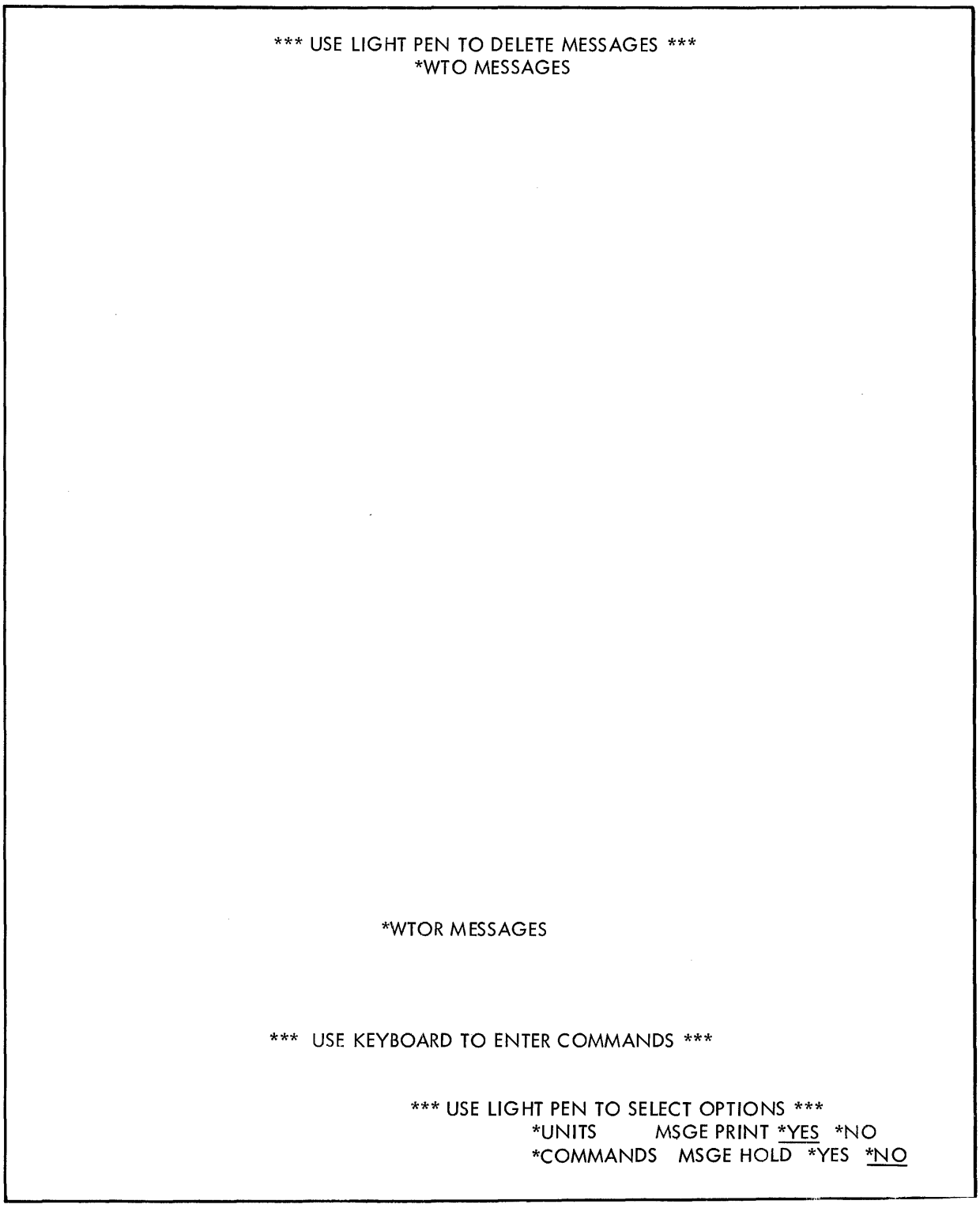


Figure 7. Basic Display

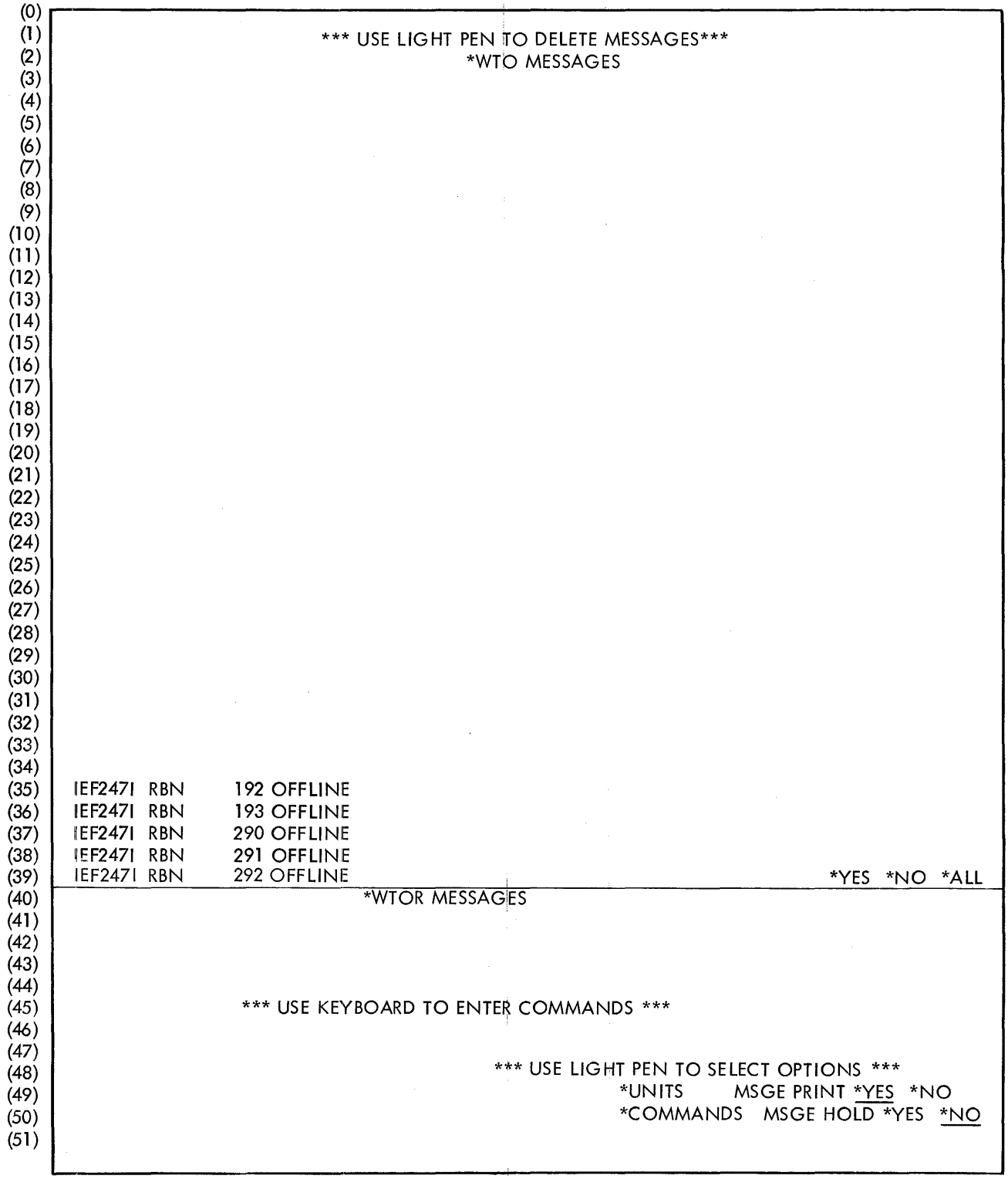


Figure 8. Basic Display With Action Option

```
(0)
(1)                                  *** USE LIGHT PEN TO DELETE MESSAGES ***
(2)                                              *WTO MESSAGES
(3)  START WTR OOE
(4)  IEF4031   WTR STARTED
(5)  IEF2361   ALLOC FOR WTR
(6)  IEF2371   IEFRDER ON 00E
(7)  IEF000A   INT REQ, 00E, **, 0200, 4000
(8)                                  *** USE LIGHT PEN TO ERASE UNITS ***
(9)  *UNIT STATUS UNIT STATUS UNIT STATUS UNIT STATUS UNIT STATUS UNIT STATUS
(10)     000 ONLINE
(11)     009 CONSOLE
(12)     00C ONLINE
(13)     00D ONLINE
(14)     00E ALLOC
(15)     016 CONSOLE
(16)     180 ALLOC
(17)     181 ALLOC
(18)     190 SYSRES
(19)     191 OFFLINE
(20)     192 OFFLINE
(21)     193 OFFLINE
(22)     280 ALLOC
(23)     281 ALLOC
(24)     290 OFFLINE
(25)     291 OFFLINE
(26)
(27)
(28)
(29)
(30)
(31)
(32)
(33)
(34)
(35)  IEF2471  RBN                    192 OFFLINE
(36)  IEF2471  RBN                    193 OFFLINE
(37)  IEF2471  RBN                    290 OFFLINE
(38)  IEF2471  RBN                    291 OFFLINE
(39)  00   IEF2471  REPLY 3 CHAR DEVICE NAME OR CANCEL
(40)                                          *WTOR MESSAGES
(41)  00   IEF2471  REPLY 3 CHAR DEVICE NAME OR CANCEL
(42)
(43)
(44)
(45)                                  *** USE KEYBOARD TO ENTER COMMANDS ***
(46)  REPLY 00, 'CANCEL'
(47)
(48)                                  *** USE LIGHT PEN TO SELECT OPTIONS ***
(49)  WTO WAITING-DELETE OR 'HOLD=NO'          *UNITS          MSGE PRINT *YES  *NO
(50)                                           *COMMANDS       MSGE HOLD *YES  *NO
(51)
```

Figure 9. Basic Display With Unit Status and Warning Message

Entry Message Display Area

The Entry message display area is used to enter and display operator commands and to enter replies to WTOR messages. The header line for this area is "***USE KEYBOARD TO ENTER COMMANDS***" (see Figure 7, line 45).

This area consists of two lines, each of which can contain a single reply or command. A reply or command can use up to 126 characters in the two lines starting at the left of the Entry display area (line 46).

You can enter a reply or a command by using the alphameric keyboard under cursor control. The cursor, an underscore, is not displayed unless it is moved to a position containing a displayable character, that is, one which is alphabetic, numeric, special, or blank. The cursor moves to the right one position as each character of the reply or command is entered.

If you make a typing error while entering a command, you can correct the error using the BACKSPACE key. The cursor is then visible and indicates the position of the next character to be entered.

Moving the cursor via the BACKSPACE and ADVANCE keys does not affect the command. However, the cursor marks the end of the text of the command. Thus, after you have made the correction, you must either retype the text from the point of error, or advance the cursor to one space beyond the last character of your text.

The end of your command is signaled by pressing the alternate coding key and hitting the numeric 5 key. After the entry is read, the Entry display area is blanked and the cursor is repositioned at the left of the Entry display area. The entry will be displayed in the WTO area, shortened to 74 characters.

You can cancel your command before signalling END by pressing the alternate coding key and hitting the numeric 0 key.

Control Display Area

The control display area of the screen (see Figure 7, lines 49 and 50) displays the options available to you for selecting and controlling information to be displayed. WTO and WTOR warning messages are displayed in this area as they occur (see Figure 9, line 49). The header line for this area is "***USE LIGHT PEN TO SELECT OPTIONS***" (see Figure 7, line 48).

Select an option by using the light pen on any of the displayed characters in the desired option. These options are: *UNITS, MSGE HOLD *YES *NO, and MSGE PRINT *YES *NO.

Units Option: This option produces a unit status table of 3-digit hex addresses of input/output devices in the system and their current status (see Figure 9, lines 10 through 25). A maximum of 96 device addresses can be displayed. The header line for the unit status table is "*UNIT STATUS UNIT STATUS..." (see Figure 9, line 9).

The address of each device is shown under the UNIT heading and the status of the device, that is, ALLOC (online and allocated), ONLINE (online and not allocated), OFFLINE, SYSRES (system residence), or CONSOLE is shown under the STATUS heading.

The unit status table is displayed in the WTO area. The table may temporarily overlay WTO messages in the area. The presence of the unit status table does not affect the adding of messages to the WTO area.

You can delete the table by using the light pen on any character in the line "***USE LIGHT PEN TO ERASE UNITS***" (see Figure 9, line 8). The messages displaced by the table are restored.

MSGE Hold Option: You can use this option to temporarily prevent additional messages from being entered in the WTO area. When the MSGE HOLD *YES option is selected by using the light pen, further WTO messages are held but not displayed. The YES is underlined when in effect. You can still delete messages using the light pen while the MSGE HOLD option is YES.

When you select the MSGE HOLD *NO option by using the light pen, further WTO messages are displayed. The NO is underlined when in effect. Any WTO message that was held by a previous selection of the MSGE HOLD *YES option is displayed, also. When the system is started the MSGE HOLD option is NO.

MSGE Print Option: You can use this option to provide hard copy output of messages and commands on an appropriate alternate console (that is, a typewriter or printer) if one is available.

Hard copy is provided when the MSGE PRINT *YES option is selected by using the

light pen on the *YES. The YES is underlined when in effect. If the MSGE PRINT *NO option is selected by using the light pen on the *NO, the hard copy output is bypassed. The NO is underlined when in effect. When the system is started, the MSGE PRINT option is YES.

ERROR HANDLING

Both synchronous and asynchronous errors can occur on the 2250. Synchronous errors can occur during an input/output operation. Asynchronous errors can occur at any time during display generation, but not while an input/output operation is being performed.

Synchronous Errors

Although the system automatically retries all synchronous errors, unrecoverable synchronous errors on either the primary or alternate console disable that device and prevent its use as an operator's console. If an unrecoverable synchronous error occurs on the 2250 when it is the primary console, the screen on the 2250 is blanked. If an alternate console is available, the system bypasses the 2250 and continues message output using the hardcopy alternate console.

If an alternate console is not available, the system places the status bytes of the CSW and two sense bytes in register 15 before putting the CPU into wait state. The sense bytes are placed in the high-order two bytes of the general register, the two status bytes in the two low order bytes.

Asynchronous Errors

When an asynchronous error occurs on the 2250, the screen is blanked and the following message appears:

"2250 ERROR HAS OCCURRED. RECENT LIGHT PEN OR KEYBOARD ACTION MAY NEED TO BE REPEATED. PLEASE USE LIGHT PEN HERE * TO RESTART DISPLAY. BUFFER ADDRESS CF ERROR IS (XXXX)"

The message that notifies you of the occurrence of an asynchronous error contains the buffer address of the byte which caused the error. If the error occurs twice consecutively at the same buffer address, the system either: (a) bypasses the 2250 and continues message output using the hard-copy console, if one is available; or (b) places the buffer address and hexadecimal "0800", indicating a data check, in register 15 before putting the CPU into wait state. The indication of a data check is placed in the high-order two bytes of the general register, while the buffer address is placed in the low-order two bytes.

When the light pen is used on any character in the message, the original display will be restored. The restored display includes WTO, WTCR and warning messages, and MSGE HOLD and MSGE PRINT options. However, unit status or message delete options (YES-NO-AIL) are not restored, and the entry display area remains blank.

SPECIAL OPERATING TECHNIQUES FCR THE 2250 OPERATCR'S CCNSCLE

With the exception of a few operating techniques peculiar to the 2250 display unit, control of the operating system is the same as from the 1052 Printer-Keyboard. Commands and replies to system messages are typed in on the keyboard shown in Figure 10, and messages to the operator are displayed on the 2250 screen.

The primary differences are that there is no REQUEST key (it is not needed), and that in place of signaling ECB by holding down the alternate coding key and hitting the numeric 5 key, you signal END using the same keys. All other differences in operating techniques are related to the display. These techniques are detailed in the following paragraphs.

How to Start the System

This procedure is the same except that while the nucleus is being readied by the nucleus initialization program (NIP), the 2250 system operator's console:

- Automatically deletes old messages when the screen is full and displays new messages.
- Considers the use of the light pen as a permanent error. The screen is blanked and another IPL is required.

How to Enter Commands from the 2250 Keyboard

- Type the command.
- Signal END by holding down the alternate coding key and hitting the numeric 5 key.

How to Correct Errors

To correct errors, you can cancel the entire line by holding down the alternate coding key and hitting the numeric 0 key and then retyping the command.

If the entire line does not need to be canceled, the errors may be corrected by backspacing. Press the BACKSPACE key and, after proper repositioning, retype the desired entry. Signal END by holding down the alternate coding key and hitting the numeric 5 key.

CAUTION: After signaling END or CANCEL, the light pen should not be used until the entry display area has been blanked; otherwise a previous light pen indication will be lost.

How to Delete Messages

- Point the light pen at the desired message in the WTO area.
- Press the foot pedal.
- Point the light pen at the desired action option to the right of the message.
- Press the foot pedal.

Note: Complete the above steps before taking other action.

How to Select MSGE Hold and Print Options

- Point the light pen at the desired option in the control display area.
- Press the foot pedal.

How to Display Unit Status

- Point the light pen at the UNITS option in the control display area.
- Press the foot pedal.

How to Delete Unit Status Display

- Point the light pen at any of the characters in the line "***USE LIGHT PEN TO ERASE UNITS***" (Figure 9, line 8).
- Press the foot pedal.

How to Shift to an Alternate Console Device

This procedure is the same as with any other console device. However, it may be necessary to perform the CANCEL function by holding down the alternate coding (ALT) key and hitting the numeric 0 key to get a display after switching the 2250 from an alternate console status to a primary console status.

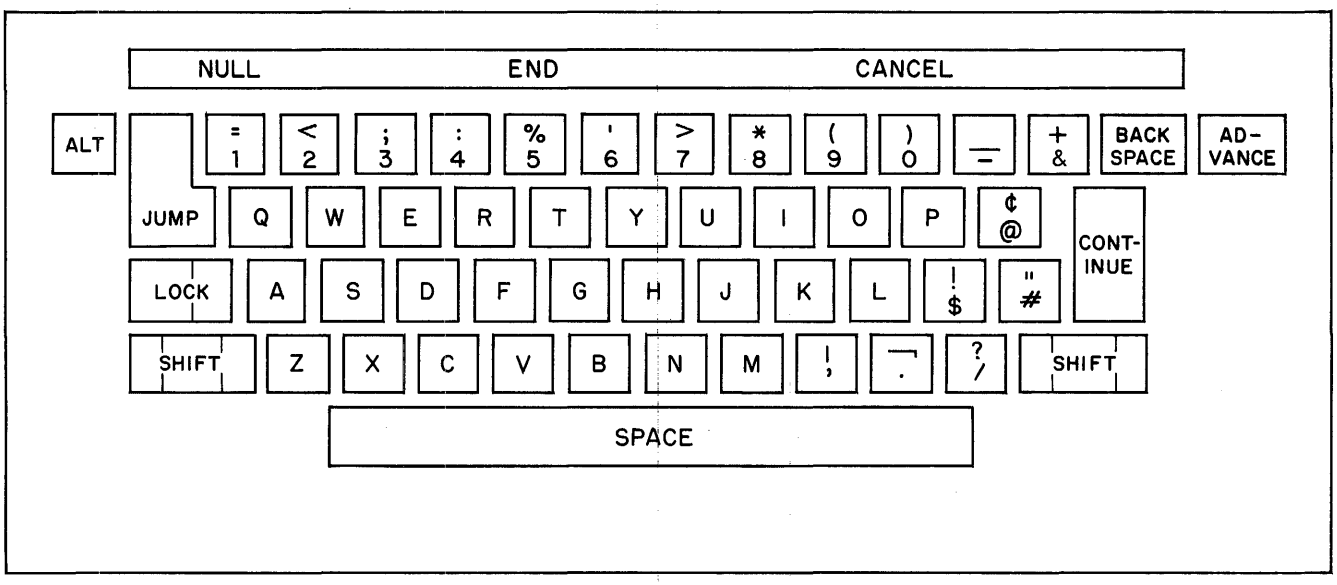


Figure 10. IBM 2250 Model 1 Keyboard

CHAPTER 4: REMOTE JOB ENTRY IN SYSTEMS WITH MVT

The remote job entry (RJE) facility of the operating system provides a way of entering jobs submitted from remote work stations into the stream.

Once a job has been entered into the job stream by RJE, execution of the job proceeds under the control of operating system scheduler routines.

All data sets created by the job are handled by the operating system data management routines. Output data sets created by remotely submitted jobs and to be returned to this remote user are placed in a distinct output class. These data sets are then removed from this output class and returned to the remote user under the direction of the RJE program.

The remote user has the same batch computing facility that is available at the central installation.

RJE provides a means for efficient operation of computing facilities by equipment centralization, and also gives substantial computing power on a demand basis to locations not requiring it on a regular basis.

In addition, it allows sharing of a common body of information within a company by widely separated departments having related needs.

RJE provides fast turnaround of computer requirements for people in all parts of a company by placing the computer facilities close to the source of input with high speed communication lines.

GENERAL CONCEPTS

Remote job entry controls a flow of data and processes that data as required. Data entering from remote sources is the primary input to the RJE system.

Jobs submitted by remote users are passed to the operating system for scheduling and execution. When the output resulting from these jobs becomes available, it is returned to the user as requested -- either immediately or on command.

RJE operates in conjunction with MVT as a system task, much like a combined reader and output writer. Jobs received from the work station are scheduled for subsequent execution.

When a remotely submitted job is completed, the job output is placed in a common SYSOUT class for RJE. RJE removes the output from this class and returns it to the work stations.

Using commands, you can supervise the central system and communicate with remote users. The central commands are:

1. BRDCST - Maintain information in the broadcast data set.
2. CENOUT - Cause output from remotely submitted jobs to be written locally.
3. MSG - Send a message to a work station.
4. SHOW - Display information pertaining to RJE.
5. START - Begin RJE operation at the central installation.
6. STOP - Cease RJE operation at the central installation.
7. USERID - Add users to or delete users from the system.

STARTUP AND CLOSEDOWN

When the central system is in operation, remote work stations may begin and end RJE activities at will. When the central system ceases operation, all work stations are closed down.

CENTRAL STARTUP AND CLOSEDOWN

Central startup and closedown are achieved by the START and STOP commands. When startup is initiated by the START command, you will get a message indicating that the RJE task is in operation. When closedown is initiated by the STOP command, you will get a message indicating completion of the RJE task.

SYSTEM OVERLOAD

An overload condition results if direct access storage space at the central installation is insufficient to meet the demands of the system.

Input already received and acknowledged by RJE is not affected by an overload condition. Any input transmission causing an overload condition is ended and must be wholly resubmitted at a later time.

In each overload situation, a message is sent to both you and the work station operator indicating the particular resource depleted. If the system continually overloads, the direct access storage space allotted for the resources must be increased to reflect the peak traffic requirements of the system more realistically.

An alternate solution may be to reschedule the work load to take advantage of periods of relative inactivity.

The total system input capacity is specified by the central installation and is dependent on the following resources:

- The quantity of SYS1.SYSJOBQE space -- specified at system generation.
- The number of concurrent jobs RJE is to maintain -- specified at RJE assembly by your system programmer.
- The quantity of space for remotely submitted SYSIN data -- specified in an RJE cataloged procedure referred to in the START command for RJE.

SYS1.SYSJOBQE depletion results when job input submitted both locally and from attached work stations exceeds the limit specified by the central installation. If this condition continues to occur, the size of the SYS1.SYSJOBQE must be increased to reflect the requirements of the system--both local and remote. This requires that SYS1.SYSJOBQE be scratched and reallocated.

An overload condition also occurs when the number of remote jobs resident in the central system exceeds the limit specified when the RJE program was assembled.

Remote jobs have residence until the output is removed from the RJE SYSOUT class. This condition is relieved by requesting the output of completed remote jobs in the system. You have this facility with the CENOUT command. If the condition continues to occur, the number of remote jobs RJE can maintain must be increased. This requires an RJE assembly by your system programmer.

Depletion of SYSIN space is the final cause of a system overload. In its cataloged RJE procedure, the installation specifies SYSIN data sets on a communication line basis. In this procedure, the installation specifies the direct access type, the volume serial number to be used for the SYSIN data sets for this line, the blocking factor for SYSIN data sets, and the maximum space available for any one input data set.

Specifying the maximum amount of space allocated for one input data set prevents one job from getting all the SYSIN space. This is a system protection feature, and no special action is necessary if a job exceeds this limit.

On the other hand, a regular depletion of the total SYSIN allocation necessitates a new catalogued procedure, referred to at START RJE time, which makes more SYSIN space available to the system.

OPERATOR AWARENESS

A message is displayed whenever an unrecoverable communication error occurs. In addition, error counts for each line are displayed.

The system keeps for each line an error count for data check, for intervention required, and for non-text time-out, and also records the number of transmissions occurring on the line.

The installation can specify threshold values for these counters when the RJE support is assembled.

If any one of the three error counters reaches its threshold count before the transmission threshold count is reached, a message is displayed. This message identifies the line and gives all three error counts and the transmission count. After the message is displayed, the error counts and the transmission count are added to accumulators, and the counters reset to zero.

You can display the value of the accumulators at any time with the SHOW IERB command. The accumulators for one line or for every support line can be requested. These accumulators are reset to zero each time the central system starts up.

Note: If the transmission count reaches its threshold count before any error count, the counters are added to the accumulators and reset to zero. No message is displayed.

RJE RESTART

RJE must be restarted whenever the operating system is restarted because of an unrecoverable error. The procedure

followed to restart RJE at the central system includes three steps.

1. The condition causing the error is corrected.

2. The operator re-IPLs the system.

3. The operator issues the START command for RJE.

These three steps restart the RJE support in the central system. RJE provides the necessary information to the work stations to insure that no information is lost.

RJE COMMANDS

You communicate with the RJE system through RJE commands. These commands provide the additional capabilities needed to control and maintain the RJE application and to communicate with RJE users and work stations. If the RJE task is not active when the commands are issued, the commands are not accepted and a message is issued.

The restrictions imposed on format and placement of these commands are identical to those for JCL command statements; that is, the commands are introduced from the keyboard-printer or the system input device.

When entered from the system input device, the commands contain the JCL identifier (//) in the first two positions of the command statement.

The commands available to the central RJE operator provide a number of capabilities.

1. Commands used to control the RJE application:

   START
   STOP

2. Commands used to maintain the RJE application:

   USERID
   CENOUT
   SHOW

3. Commands used to communicate in the RJE system:

   MSG
   BRDCST

An RJE command statement cannot be continued. It must be coded on one card or card image. No abbreviation of these commands (except START and STOP) is allowed.

RJE will buffer central commands, up to 100 (defined in the assembly of RJE), so that the operator can have more than one command pending at a time. All buffered commands will be processed.

BRDCST -- Maintain the RJE Broadcast Messages

The BRDCST command is used to maintain broadcast messages which are sent to work stations on request and when the stations become active. The messages give such data as when the central station is closing down, and when a central resource is added or deleted.

These messages are kept in a data set, on a direct access device, with provisions for up to 100 broadcast messages. Each broadcast message in the data set is numbered to correspond with a numbered slot. These slots are numbered from 0 to 99 and are either active (containing a message), or inactive (containing no message).

The BRDCST command lets you:

- Insert a new message.
- Add a new message.
- Change an existing message.
- Remove an existing message.
- Collect active messages into the low-numbered slots.
- Clear the data set of all existing messages.

You can get a copy of the active broadcast messages through the SHOW BRDCST command.

Message text must be entered in upper case.

Operation	Operand
BRDCST	nn,'text' 'text' nn DELETE Ann,'text' PACK

nn, 'text'
specifies the 'text' be placed into the record of the data set associated with slot number 'nn'. The slot is set to active regardless of the original status of the specified slot.

'text'
specifies that text is to be placed into the lowest numbered inactive slot. If no slots are inactive, the command is refused.

nn
specifies that the text contents of slot number 'nn' be deleted. This is done by merely setting that slot to inactive.

DELETE
requests the deletion of all texts for the BRDCST data set.

Ann, 'text'
specifies the insertion of the TEXT into slot number 'nn' after moving sequentially the texts of that slot and of all consecutive active slots up into a higher numbered inactive slot. This preserves the contents and sequential order of all original texts. If there is no high inactive slot to receive the pushed up texts (or if slot 99 is specified), this command is refused.

PACK
specifies the collection of all active slots at the 00 end of the directory and of all inactive slots at the 99 end. This function may be used to recover from the insert command (see previous paragraph) being refused due to a lack of a higher inactive slot. The contents and sequential order of all active slots are preserved.

CENOUT -- Give RJE Output to Local Output Writers

The CENOUT command is used to remove job output in the RJE SYSOUT class and process it with the central installation output writers.

This command allows you to retrieve output of completed remotely-submitted jobs which cannot be transmitted or has not been requested by an RJE user.

The RJE system places the output data sets and system messages in the originally-specified SYSOUT class. The disposition of the output is the same as that of any other data for the SYSOUT class at the central installation.

Only output of completed jobs is placed in the originally-specified SYSOUT classes. Jobs completing after the CENOUT command has been processed remain in the RJE SYSOUT class.

Operation	Operand
CENOUT	J=jobname

J=jobname
specifies the name of the job whose output is to be handled by the central installation output writers. If the job whose output is requested is not complete or the job is not in the RJE system, the command is rejected and a message is issued.

MSG -- Communicate with RJE Users

The MSG command is used to send messages to the users and work stations constituting the RJE system. You can selectively route a message to:

- A specific user currently logged on.
- A specific work station.
- A specific user or, if the user is not logged on a specific station.
- All work stations logically attached to the RJE system.

In addition, the MSG command lets you remove from the system messages whose transmission is waiting for a work station startup. This option is normally used either when communication between the central installation and a work station is not possible due to some failure, or when the data set where these pending messages are kept becomes full.

Operation	Operand
MSG	{[U=userid,] [T=termid,] M='text' {D=termid

U=userid
specifies that the message is to be sent to the user identified by a userid which consists of three alphameric characters -- for example, U2T or 123. The message is sent to the user if he is logged on. If he is not logged on and the T keyword is omitted, the message is not sent. A response indicating this condition is returned to the operator. If both T and U keyword parameters are specified and the user is not logged on, the message is either sent to the work station or held until work station startup.

T=termid
specifies that the message is to be sent to the work station identified by a termid which consists of one to eight alphameric characters, the first of which must be alphabetic -- for example, RALEIGH or TERM23. If the work station is inactive, the message waits for the work station to start up.

M='text'
specifies the text of the message which is to be sent. The message text must be framed with apostrophes and can include as many as 40 printable characters and blanks. All text characters must be upper-case. Any apostrophes included as part of the text must be paired; each pair counts as one text character. If you omit the U and T keyword parameters, the message is sent to all work stations logically attached to the RJE system. Work stations inactive when the command is issued do not receive the message.

D=termid
specifies the deletion of pending messages for the work station identified by termid. A copy of these messages can be obtained with a SHOW MSGS, termid command before entering the MSG D=termid command.

SHOW -- Display RJE Information

The SHOW command is used to request a keyboard-printer display of RJE information.

The information desired is specified with a coded value in the SHOW command operand. Only one uppercase parameter can be specified for the operand of each command.

If more than one type of available RJE information is wanted, each type must be requested with a separate SHOW command.

Operation	Operand
SHOW	JOBS TERMS USERS DEFER DEFER,userid BRDCST MSGS MSGS,termid LERB LERB,linename

JOBS
specifies a request for a list of RJE jobs at the central installation and their status. The status returned indicates only whether the job is complete.

TERMS
specifies a request for a list of the work stations in the RJE system and the state of each work station. The display indicates whether the work station is currently attached to the system.

USERS
specifies a request for a list of all valid userids and their associated keys with an indication of whether or not each user is currently logged on.

DEFER
specifies a request for a list of all jobs which have deferred job output pending in the RJE SYSOUT class, with an indication for each job of the time the output has remained in the system. The indication returned is the number of central closedowns occurring since the output was created.

DEFER,userid
specifies a request for a list of all jobs associated with the user specified by userid which have deferred output pending in the RJE SYSCUT class, with an indication of the time each output has remained in the system. The indication returned is the number of central closedowns occurring since the output was created.

BRDCST
specifies a request for a copy of the current broadcast messages.

MSGS
specifies a request for a copy of all RJE messages whose transmission is waiting for a work station to start up. The work station to which each message is directed is indicated in the list.

MSGS,termid
specifies a request for a copy of RJE messages pending for the work station specified by termid. If termid does not correspond to a work station in the system, the command is rejected.

LERB
specifies a request for a list of the current values of all line error accumulators for all communication lines being used for RJE. The list indicates for each line the three error counter values (data check, non-text time-out, and intervention required) and the transmission counter values. These values are cumulative values since the last RJE central startup. They are reset to zero at central startup.

LERB,linename
specifies a request for the error and transmission counts for a particular line. The linename is the name which was specified for the line when the entral RJE program was assembled. You will receive a message containing the three error counter values and the transmission counter value for the line designated.

START -- Start RJE System Process

The START command is used to start operation of RJE at the central installation.

Operation	Operand
{START / S}	procname,,,{FORM / NFMT / NONE}

Note: The two commas following the delimiting comma after procname are required to indicate the absence of the parameters volumeserial and parmvalue. See the description of the START command in Chapter 1, 2, or 3.

procname
specifies the name of the cataloged procedure for remote job entry operation. This procedure name must begin with characters RJE.

FORM
specifies that RJE is to start from scratch. This form is to be coded only if the operating system has been loaded with the following IPL option since RJE closed down: in the SET command, 'F' was specified in the Q keyword subparameter list. 'F' indicates that the job queue data set is to be formatted during the IPL. The operating system formatting of the job queue data set deletes all jobs within the RJE System. The FORM parameter removes all references to jobs in the RJE job table. If FORM is specified, all jobs within the RJE System are deleted regardless of the operating system startup.

NFMT
specifies that RJE is to restart. This form is to be coded only if the operating system has been loaded since RJE closed down and the subparameter 'F' of the keyword Q in the SET command has not been specified. If the operator reloads the operating system more than once since RJE closed down and during any of the IPL procedures specified 'F' in the SET command, then FORM must be coded on the START statement for RJE and not NFMT. If the operator loads specifying Q=([unitname] [,F]) in the SET command and RJE job table will retain its references to the jobs deleted by reformatting the job queue data set. These references can be removed only by specifying FORM. If the operator has not loaded the operating system since RJE closed down and NFMT is specified, no job output existing prior to closedown can be retrieved during this execution of RJE. To recover the output the operator must reload (do not specify 'F' in the SET command) and start RJE specifying NFMT.

NCNE
specified when the operating system has not been loaded since RJE closed down. If None is specified on the START statement and the operator had loaded the operating system since RJE closedown, all jobs within the RJE System are deleted. In addition, if the operator specified 'F' in the SET command during the IPL, the RJE job table will retain its references to the deleted jobs. Any attempts to access the deleted jobs will cause unpredictable table results. The RJE job table references can be freed only by specifying FORM.

STOP -- Stop RJE System Process

The STOP command is used to stop operation of RJE at the central installation.

Operation	Operand
{STOP / P}	procname

procname
specifies the name of the cataloged procedure for remote job entry operations. This procedure name is the same as that specified in the START command.

USERID -- Modify the RJE User Directory

The USERID command is used to modify and maintain the RJE user directory. The USERID command lets you:

- Add a userid/key pair to the user directory.
- Remove a userid/key pair from the user directory.

These facilities are provided dynamically and don't require a reassembly of the RJE program.

Each userid in the directory must be unique.

If you submit a command to add a userid that is currently contained in the user directory, RJE rejects the command and returns a message indicating that the userid is already contained in the directory.

When a userid is removed from the directory, all jobs currently in the system associated with that userid are also removed.

Operation	Operand
USERID	userid,key,{ADD / DELETE}

userid
specifies the userid which is to be added or removed from the user directory. If this userid is already in the user directory, RJE rejects the command and sends you a message. The userid consists of three alphameric characters -- for example, U2T or 123.

key
specifies the protection key assigned to the userid designated in the user parameter. The same key may be assigned to several userids. The key consists of three alphameric characters -- for example, 123 or U51.

ADD
specifies that the userid and key are to be added to the user directory. If the addition of a userid and key is desired and no space is available in the user directory, the RJE program must be reassembled to increase the size of the user directory.

DELETE
specifies that the userid and key are to be deleted from the user directory.

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.
START key	Starts instruction execution. (Works only if the CPU is in the stopped mode.)
STOP key	Puts the CPU in the stopped mode.
SYSTEM RESET key	Stops instruction processing, and resets the CPU, channels, online nonshared control units, and I/O devices. Does not reset registers.
CHECK RESET key	Resets the console check lights.
INTERRUPT key	Stops program execution by causing an external interruption. Allows you to switch from the primary console to the alternate console.
DISPLAY key	Displays information you specify by setting appropriate panel switches. Works only if the CPU is in the stopped mode.
ADDRESS COMPARE switch	Stops the CPU when it reaches any address you select in advance. Used to load a secondary nucleus at IPL time.
LOAD UNIT switches	Tell the system where to get the IPL program when you hit the LOAD key.
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.
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
LOAD light	Goes on when the LOAD key is pushed, goes off when the IPL record is loaded.
WAIT light	Goes on when the CPU is in the wait state.
MANUAL light	Goes on when the CPU is in the stopped mode.
TEST light	Goes on when the switches with a NORMAL position are not pointing to that position.
SYSTEM light	Goes on when the CPU or channels are running.
STORAGE DATA lights	Go on after the DISPLAY key is hit when their corresponding keys are set to specify data to be loaded into storage or into registers.
INSTRUCTION COUNTER/STORAGE ADDRESS lights	Go on after the DISPLAY key is hit when their corresponding keys are set to specify the storage address where the CPU has stopped.

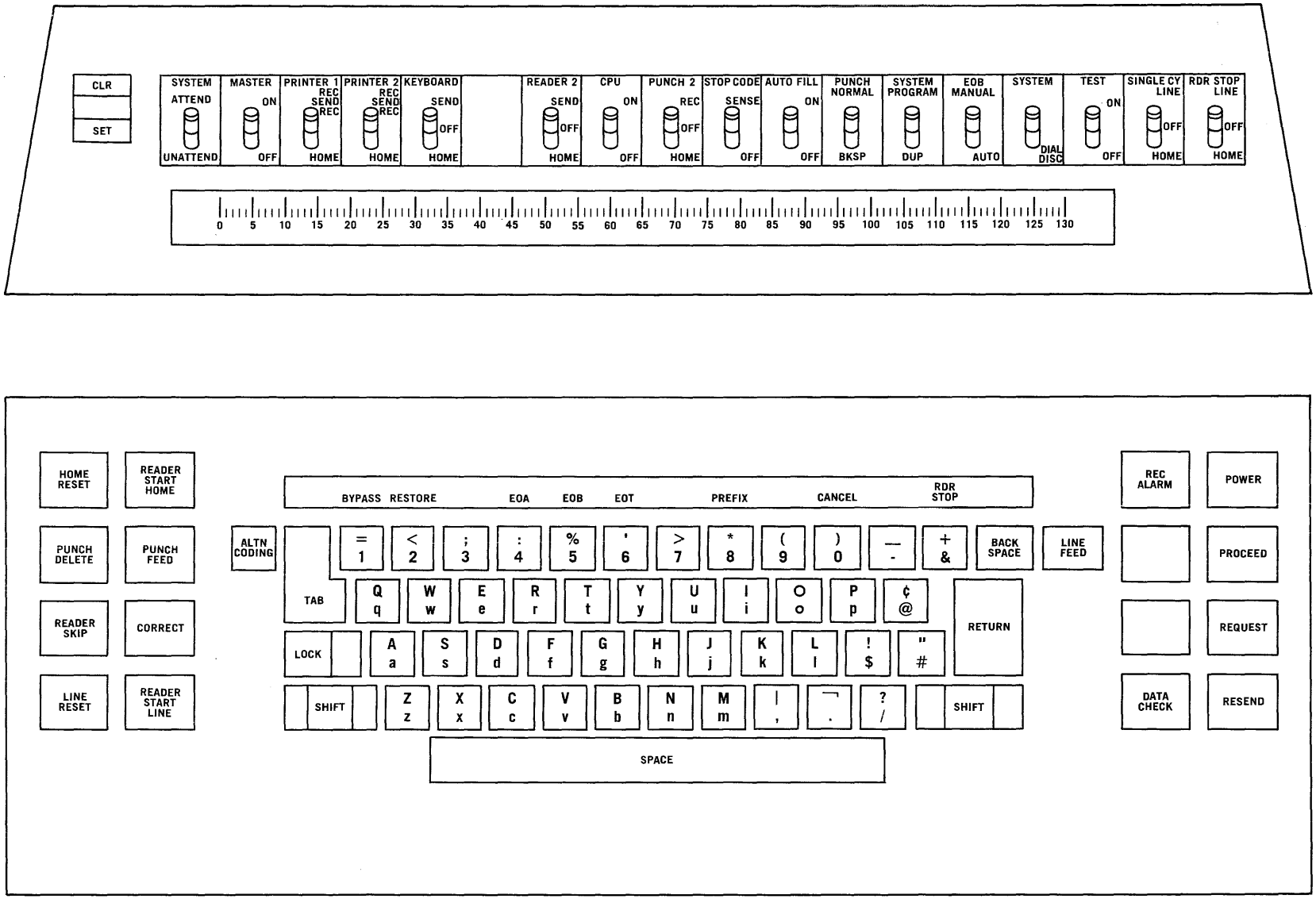


Figure 11. Model 3 Keyboard and Switch Panel

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

- Hit 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 hitting the numeric 5 key.

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

To correct errors, cancel the entire line by holding down the alternate coding key and hitting 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 11, 12, 13, 14, and 15.

HOW TO ENTER A COMMAND THROUGH A COMPOSITE CONSOLE

- Hit 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.
- Hit the START and EOF 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.
- Hit 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 TC 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 MCD option, the RAM option, the RQ option, the RSVC option, the SQS option, the MIN option, the QBF option, the MPS option, the TMSL option, or some combination of these options.

If no changes are to be made, issue REPLY id,'U', or simply signal EOB.

BLDL CPTICN

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.

MIN CPTION

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.

MOD CPTION

This option specifies to the optional machine check handler (MCH) and channel check handler (CCH) the model number of the IBM System/360 on which the system is running. To alter this option, issue REPLY id,'MOD=xx' where xx is the two-digit model number. The default value is 65.

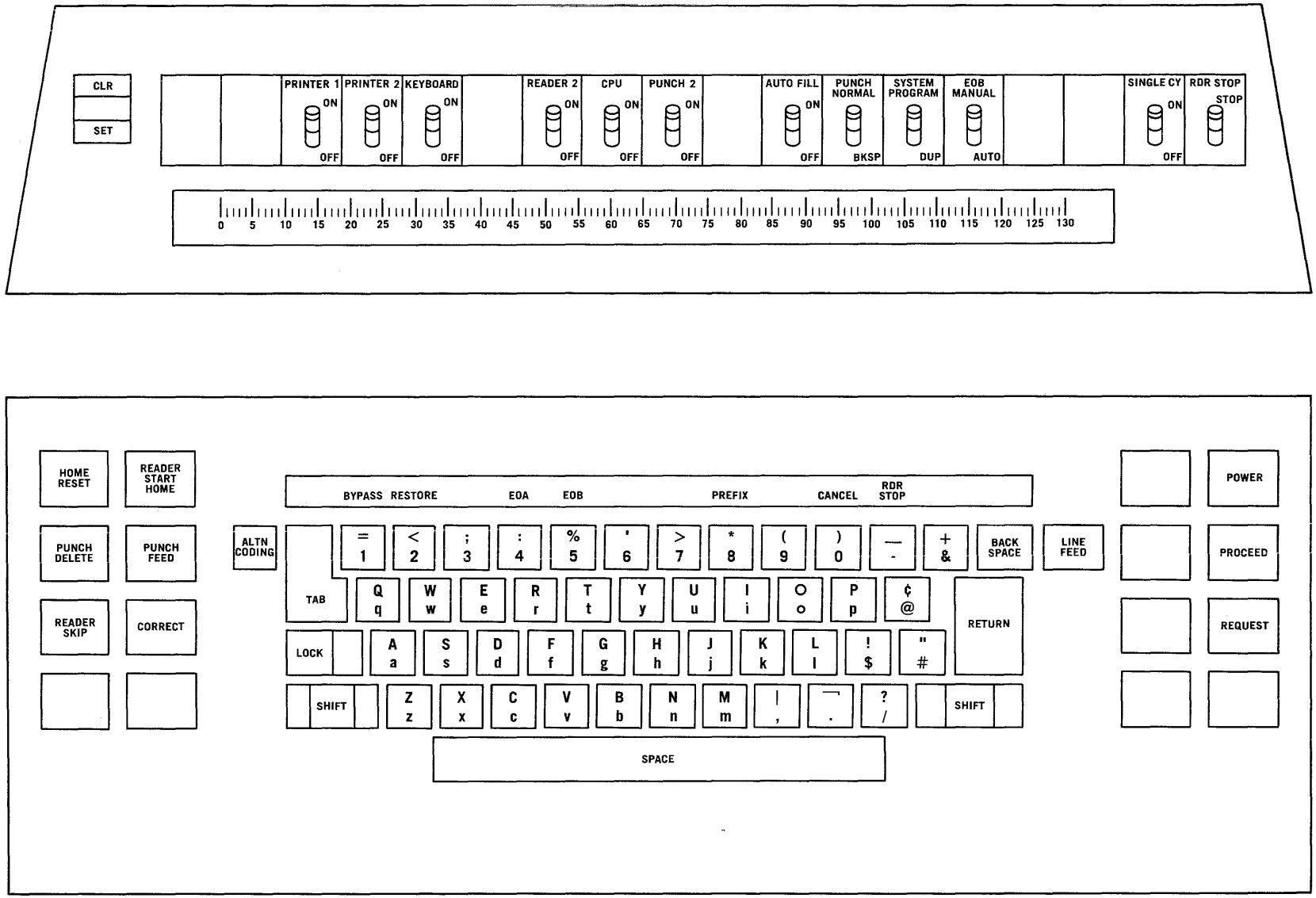


Figure 12. Model 5 Keyboard and Switch Panel

If MOD=65 is specified or assumed and if MCH and CCH are present in the system, they will perform recovery processing. However, if the CPU is not really a 65, the MCH and CCH results will be invalid and unpredictable. If MCH and CCH are present but the number specified is for a model other than 65, the system is put in a wait state when a machine check or channel check interruption occurs.

MPS OPTION

This option specifies to the system that it is to alter the size of the system-generated master scheduler region. To alter the size specified at system generation, issue REPLY id,'MPS=n' where n is the number of 2048-byte blocks of storage to be reserved for the master scheduler region.

The value for n is supplied by your system programmer.

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.

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 or MFT. 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 (multiple lists are supported only in systems with MVT).

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. In a system with MFT, the parameter is used to override the size specified at system generation, and n is the number of bytes of storage to be used for system queue space. The minimum value is 1600; this value is assumed as a default if less than 1600 bytes are requested in the reply.

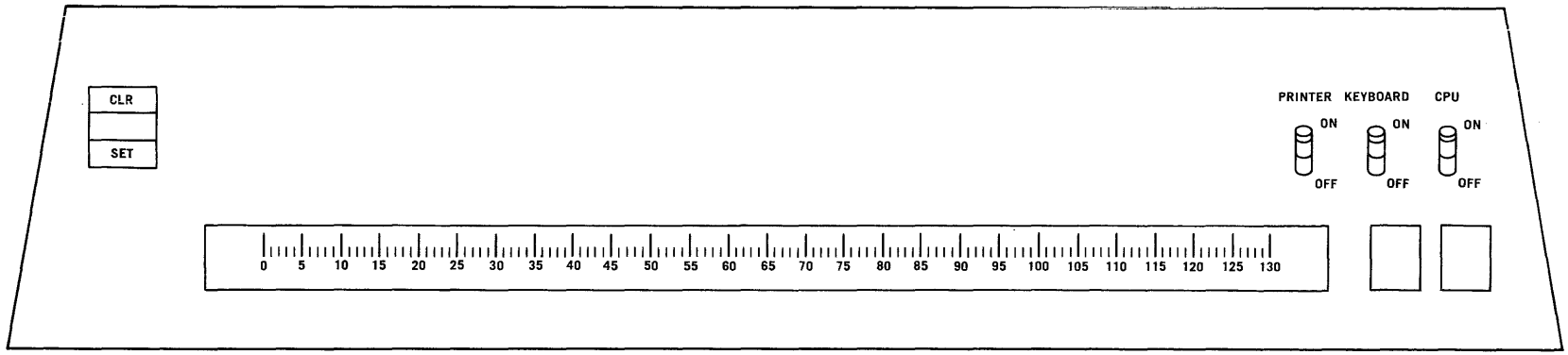
TMSL OPTION

This option specifies that the system is to include time slicing -- the revolving assignment of CPU time. The supervisor limits the amount of time each task in a time-sliced group receives at any one time. After the tasks have all had their slice, the sequence is repeated again and again.

Not all tasks that the system is performing need be time-sliced. If some are not, they are given control according to priority, competing with the time-sliced group as a whole for CPU time.

In MFT, there is one time-sliced group and it relates to a set of adjacent partitions, while in MVT there can be more than one group, each group relating to the tasks within a specified priority.

In systems with MFT, either during initialization or by issuing a DEFINE command, you can alter the time-slicing specification made when the system was generated. You can change the time-slice or the partitions in the time-sliced group, or both; you can cancel time slicing or redefine it via the DEFINE command. Examples of how to do this are given in Chapter 2 under the heading Entering the Partition Definitions.



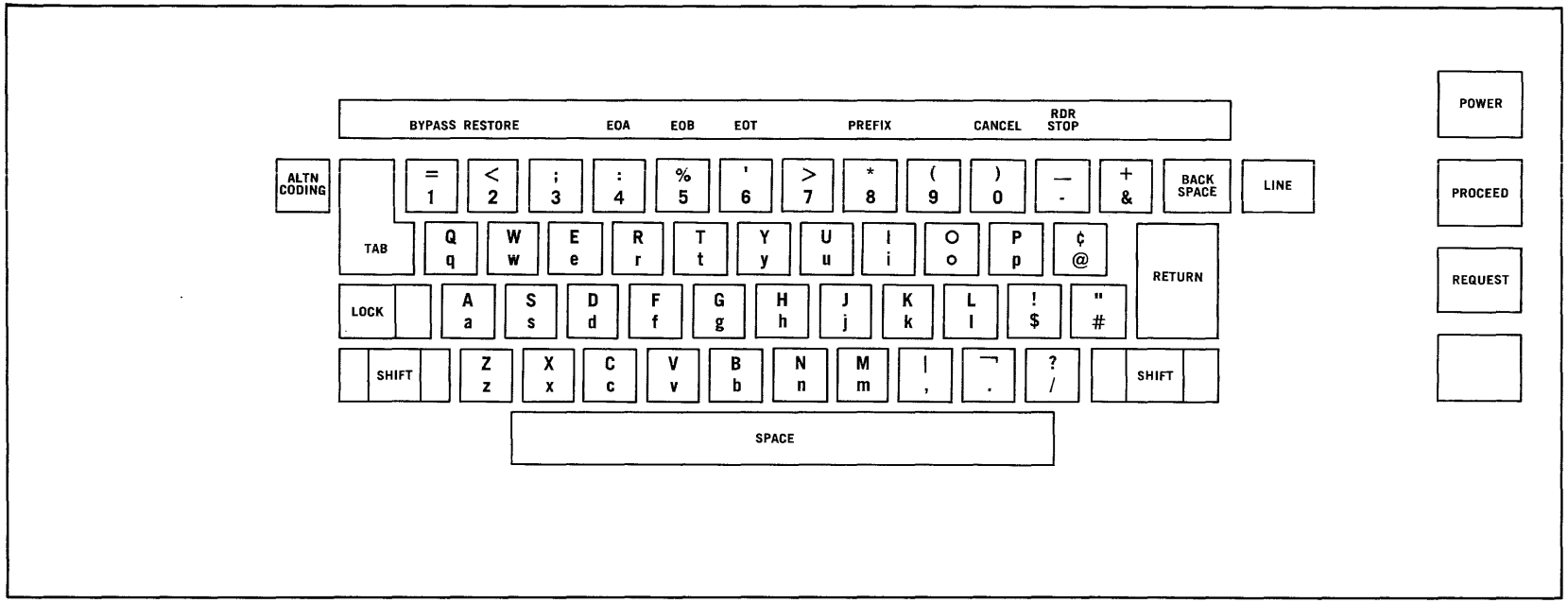


Figure 13. Model 6 Keyboard and Switch Panel

In systems with MVT, during nucleus initialization, you can change the time slice and the job priorities to be time-sliced, or you can cancel time-slicing for this IPL. But remember that you cannot specify more time-slicing groups than were specified at system generation (if you try to, you will get an error message) and that you must specify all time-slicing specifications that you intend to have in effect if you specify any. That is, if three time-sliced groups were specified at system generation and you only changed two of them at nucleus initialization, the third would be automatically canceled. If you tried to specify four in this case, you would get an error message.

To alter the time-slicing specification in systems with MVT, enter:

```
REPLY id,'TMSL=(priority,value)
        ,(priority,value)...'
```

where

priority is a number from 0 through 13 designating that tasks with that priority number are to belong to a time-sliced group, and

value is a number, 20 or larger, up to four digits, designating the maximum number of milliseconds each task in the time-sliced group can have control of the CPU at any one time.

You must specify a separate priority and value for each time-sliced group you want to have in the system.

To cancel time slicing, enter:

```
REPLY id,'TMSL=,'
```

ALTERING OR CANCELING AN OPTION

When altering or canceling an option, you can key in the name of the option in upper or lower case letters.

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, and change parameters. 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.

You must always use a comma as part of a parameter that cancels an option. When altering several options, you can separate the parameters with either a comma or a blank, but a comma is the preferred separator.

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'
```

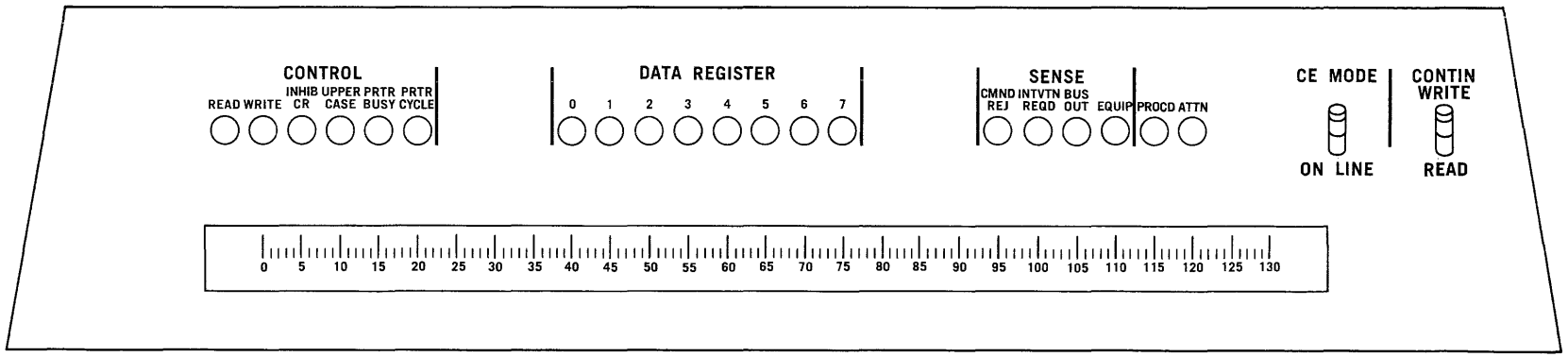
HOW TO SPECIFY JOB QUEUE PARAMETERS (MFT AND MVT SYSTEMS)

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.



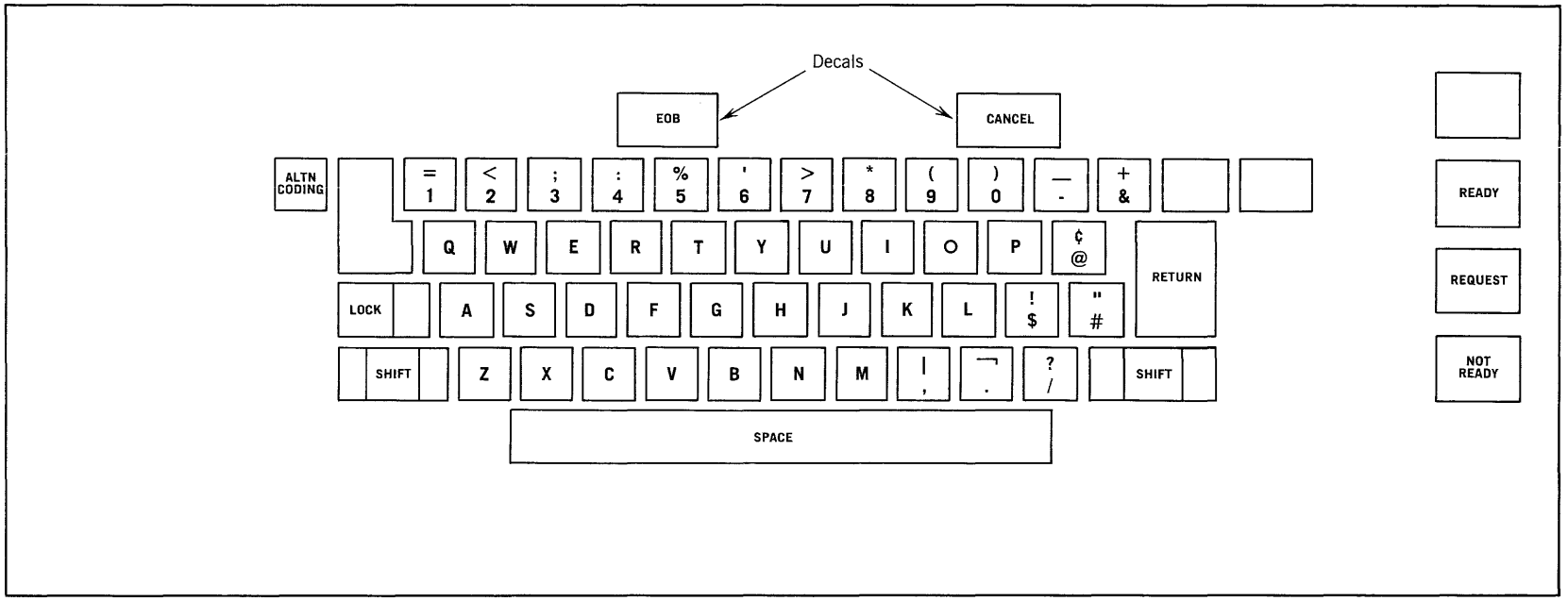


Figure 14. Model 7 Keyboard and Switch Panel

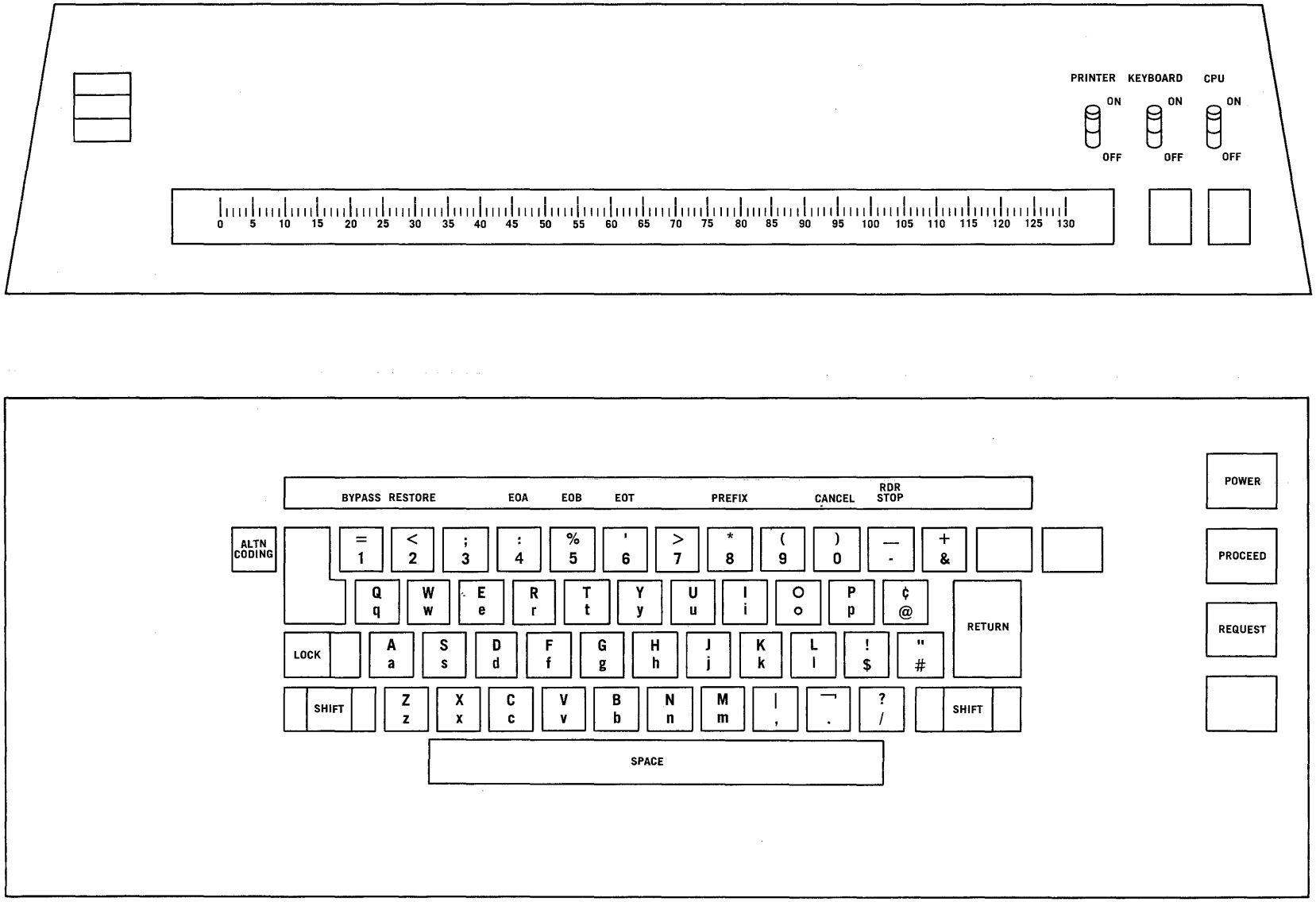


Figure 15. Model 8 Keyboard and Switch Panel

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 REPLY TO UCS MESSAGES

The Universal Character Set (UCS) special feature provides for printing any set of up to 240 graphics by the 1403 printers attached to the 2821 control unit.

The 2821 has one 240 character read/write storage unit (buffer) for each printer. Each buffer position corresponds to a character on the chain or train of the respective printer.

The one-to-four byte code (character set-code) identifies the character set image in the system library that is to be loaded by the operating system into the UCS buffer. It also tells you which chain or train cartridge to physically mount on the printer.

The following one through four byte identification codes correspond to the IBM standard character sets.

AN	alphameric
HN	alphameric
PCAN	alphameric[1]
PCHN	alphameric[1]
PN	alphameric (PL/I)
QNC	alphameric (PL/I-commercial) [1]
QN	alphameric (PL/I) scientific)
RN	FORTRAN-COBOL-commercial
SN	text printing[1]
TN	text printing
XN	high-speed alphameric
YN	high-speed alphameric[1]

User-designed character set images are stored in the system library and assigned a character set-code defined by the user.

All character set-code used by an installation should be familiar to programmers and operators to insure proper use of UCS.

---

[1]Preferred character set.

The FOLD option causes certain EBCDIC characters to be printed with the graphics corresponding to other EBCDIC characters.

The FOLD option should not be specified unless the programmer requests it or it is known that the character set specified requires folding. See the publication for the IBM 2821 Control Unit (A24-3312) for details.

The VERIFY option causes the character set image to be displayed on the printer. This allows you to verify that the chain or train mounted on the printer corresponds to the character set-code requested.

You must reply to UCS messages as indicated by the UCS messages shown in the publication IBM System/360 Operating System, Messages and Codes.

Caution is necessary when you substitute an alternate chain or train. It is recommended that the alternate chain or train have at least the corresponding graphics to the chain or train requested by the programmer.

The FOLD and VERIFY options must be respecified, as applicable, when substituting an alternate chain or train. If the requested chain or train is specified in the reply, the FOLD and VERIFY options specified by the programmer are honored.

The following chart illustrates a comparison of the graphics for each of the IBM standard character sets. (Sample print-outs for the standard IBM chains and trains follow the chart.)

The REPLY commands which contain the UCS parameters may be made in upper or lower case and, in general, follow the format,

REPLY id,'character set-code [,F / ,FOLD] [,V / ,VERIFY]'

where you indicate the appropriate parameters. The brackets indicate an optional parameter and are not included in the reply. Examples of the REPLY command for UCS being a character set-code of AN and different combinations of the FOLD and/or VERIFY options are:

```
REPLY id,'AN'
REPLY id,'AN,F'      (VERIFY omitted)
REPLY id,'AN,,V'     (FOLD omitted)
REPLY id,'AN,FOLD,VERIFY'
```

CHARACTER SET CODE	SOURCE CHARACTERS			
	TOTAL	ALPHA	NUMERIC	SPECIAL CHARACTERS
AN	48	A–Z	0–9	/ @ # ¤ . + * $ – % , &
HN	48	A–Z	0–9	/ ' =) . + * $ – (, &
PCAN	48	A–Z	0–9	/ @ # ¤ . + * $ – % , &
PCHN	48	A–Z	0–9	/ ' =) . + * $ – (, &
PN, QNC, and QN	60	A–Z	0–9	/ ' =) . + * $ – (, & " — : | > ? # % @ < ; ¬
RN	52	A–Z	0–9	/ ' =) . + * $ – (, & ¤ # % @
SN	84	A–Z a–z	0–9	/ ') . + * $ – (, & " — : | ¤ ? ! % @ ¢ ;
TN	120	A–Z a–z	0–9	/ ' =) . + * $ – (, & " — : | ¤ ? ! % @ ¢ ; ¬ – • ¬ ⌐ ⌟ ⌞ { } [ ] ▪ ≥ ≤ > < ≠ # °) (+ – 0 9 8 7 6 5 4 3 2 1 ± +
XN	40	A–Z	0–9	. * $,
YN	42	A–Z	0–9	# . * $ – ,

AN IMAGE

12	34	56	78	90	#@	/S	TU	VW	XY	Z&	,%	JK	LM	NO	PQ	R-	$*	AB	CD	EF	GH	I+	.⌑
12	34	56	78	90	#@	/S	TU	VW	XY	Z&	,%	JK	LM	NO	PQ	R-	$*	AB	CD	EF	GH	I+	.⌑
12	34	56	78	90	#@	/S	TU	VW	XY	Z&	,%	JK	LM	NO	PQ	R-	$*	AB	CD	EF	GH	I+	.⌑
12	34	56	78	90	#@	/S	TU	VW	XY	Z&	,%	JK	LM	NO	PQ	R-	$*	AB	CD	EF	GH	I+	.⌑
12	34	56	78	90	#@	/S	TU	VW	XY	Z&	,%	JK	LM	NO	PQ	R-	$*	AB	CD	EF	GH	I+	.⌑

HN IMAGE

12	34	56	78	90	='	/S	TU	VW	XY	Z&	,(	JK	LM	NO	PQ	R-	$*	AB	CD	EF	GH	I+	.)
12	34	56	78	90	='	/S	TU	VW	XY	Z&	,(	JK	LM	NO	PQ	R-	$*	AB	CD	EF	GH	I+	.)
12	34	56	78	90	='	/S	TU	VW	XY	Z&	,(	JK	LM	NO	PQ	R-	$*	AB	CD	EF	GH	I+	.)
12	34	56	78	90	='	/S	TU	VW	XY	Z&	,(	JK	LM	NO	PQ	R-	$*	AB	CD	EF	GH	I+	.)
12	34	56	78	90	='	/S	TU	VW	XY	Z&	,(	JK	LM	NO	PQ	R-	$*	AB	CD	EF	GH	I+	.)

PCAN IMAGE

12	34	56	78	90	,-	PQ	R#	$@	/S	TU	VW	XY	Z⌑	.*	12	34	56	78	90	,-	JK	LM	NO	AB	CD	EF	GH	I+	.*
12	34	56	78	90	,-	PQ	R&	$%	/S	TU	VW	XY	Z⌑	.*	12	34	56	78	90	,-	JK	LM	NO	AB	CD	EF	GH	I+	.*
12	34	56	78	90	,-	PQ	R#	$@	/S	TU	VW	XY	Z⌑	.*	12	34	56	78	90	,-	JK	LM	NO	AB	CD	EF	GH	I+	.*
12	34	56	78	90	,-	PQ	R&	$%	/S	TU	VW	XY	Z⌑	.*	12	34	56	78	90	,-	JK	LM	NO	AB	CD	EF	GH	I+	.*

PCHN IMAGE

12	34	56	78	90	,-	PQ	R=	$'	/S	TU	VW	XY	Z)	.*	12	34	56	78	90	,-	JK	LM	NO	AB	CD	EF	GH	I+	.*
12	34	56	78	90	,-	PQ	R&	$(	/S	TU	VW	XY	Z)	.*	12	34	56	78	90	,-	JK	LM	NO	AB	CD	EF	GH	I+	.*
12	34	56	78	90	,-	PQ	R=	$'	/S	TU	VW	XY	Z)	.*	12	34	56	78	90	,-	JK	LM	NO	AB	CD	EF	GH	I+	.*
12	34	56	78	90	,-	PQ	R&	$(	/S	TU	VW	XY	Z)	.*	12	34	56	78	90	,-	JK	LM	NO	AB	CD	EF	GH	I+	.*

PN IMAGE

12	34	56	78	90	XY	/S	TU	VW	|:	_"	,=	JK	LM	NO	PQ	R-	Z(	AB	CD	EF	GH	I+	.)	%$	*#	&@	<;	¬	?>
12	34	56	78	90	XY	/S	TU	VW	|:	_"	,=	JK	LM	NO	PQ	R-	Z(	AB	CD	EF	GH	I+	.)	%$	*#	&@	<;	¬	?>
12	34	56	78	90	XY	/S	TU	VW	|:	_"	,=	JK	LM	NO	PQ	R-	Z(	AB	CD	EF	GH	I+	.)	%$	*#	&@	<;	¬	?>
12	34	56	78	90	XY	/S	TU	VW	|:	_"	,=	JK	LM	NO	PQ	R-	Z(	AB	CD	EF	GH	I+	.)	%$	*#	&@	<;	¬	?>

QNC IMAGE

123	456	789	0#@	/ST	UVW	XYZ	&.%	JKL	MNO	PQR	-$*	ABC	DEF	GHI	",=
123	456	789	0#@	/ST	UVW	XYZ	&.%	JKL	MNO	PQR	-$*	ABC	DEF	GHI	|:_
123	456	789	0#@	/ST	UVW	XYZ	&.%	JKL	MNO	PQR	-$*	ABC	DEF	GHI	<;¬
123	456	789	0#@	/ST	UVW	XYZ	&.%	JKL	MNO	PQR	-$*	ABC	DEF	GHI	'?>
123	456	789	0#@	/ST	UVW	XYZ	&.%	JKL	MNO	PQR	-$*	ABC	DEF	GHI	)+(

QN IMAGE

12	34	56	78	90	XY	/S	TU	VW	_"	$*	, =	JK	LM	NO	PQ	R-	Z (	AB	CD	EF	GH	I+	.)
12	34	56	78	90	XY	/S	TU	VW	< ;	#*	, =	JK	LM	NO	PQ	R-	Z (	AB	CD	EF	GH	I+	.)
12	34	56	78	90	XY	/S	TU	VW	? >	@*	, =	JK	LM	NO	PQ	R-	Z (	AB	CD	EF	GH	I+	.)
12	34	56	78	90	XY	/S	TU	VW	¬ '	&*	, =	JK	LM	NO	PQ	R-	Z (	AB	CD	EF	GH	I+	.)
12	34	56	78	90	XY	/S	TU	VW	| :	%*	, =	JK	LM	NO	PQ	R-	Z (	AB	CD	EF	GH	I+	.)

RN IMAGE

12	34	56	78	90	XY	/S	TU	VW	'@	$*	, =	JK	LM	NO	PQ	R-	Z (	AB	CD	EF	GH	I+	.)
12	34	56	78	90	XY	/S	TU	VW	%@	$*	, =	JK	LM	NO	PQ	R-	Z (	AB	CD	EF	GH	I+	.)
12	34	56	78	90	XY	/S	TU	VW	#@	$*	, =	JK	LM	NO	PQ	R-	Z (	AB	CD	EF	GH	I+	.)
12	34	56	78	90	XY	/S	TU	VW	⌑@	$*	, =	JK	LM	NO	PQ	R-	Z (	AB	CD	EF	GH	I+	.)
12	34	56	78	90	XY	/S	TU	VW	&@	$*	, =	JK	LM	NO	PQ	R-	Z (	AB	CD	EF	GH	I+	.)

SN IMAGE

12	34	56	78	90	& .	/S	TU	VW	XY	Z ,	$*	JK	LM	NO	PQ	R-	" :	AB	CD	EF	GH	I+	ab	cd	ef	gh	ij	kl	mn	op	qr	st	uv	wx	yz	@'	( )	—|		
12	34	56	78	90	& .	/S	TU	VW	XY	Z ,	$*	JK	LM	NO	PQ	R-	" :	AB	CD	EF	GH	I+	ab	cd	ef	gh	ij	kl	mn	op	qr	st	uv	wx	yz	@'	( )	—|		
12	34	56	78	90	& .	/S	TU	VW	XY	Z ,	$*	JK	LM	NO	PQ	R-	" :	AB	CD	EF	GH	I+	ab	cd	ef	gh	ij	kl	mn	op	qr	st	uv	wx	yz	@'	( )	—|	? !	; ¢	%⌑

TN IMAGE

12	34	56	78	90	= .	/S	TU	VW	XY	Z ,	#&	JK	LM	NO	PQ	R-	" :	AB	CD	EF	GH	I+	ab	cd	ef	gh	ij	kl	mn
op	qr	st	uv	wx	yz	@'	? ;	± !	$*	%⌑	$^{1\,2}$	$^{3\,4}$	$^{5\,6}$	$^{7\,8}$	$^{9\,0}$	$^{-\,+}$	$^{(\,)}$	°¬	( )	≠+	> <	≤ ≥	¢ =	[ ]	} {	└┘	┌┐	° _	—|
12	34	56	78	90	= .	/S	TU	VW	XY	Z ,	#&	JK	LM	NO	PQ	R-	" :	AB	CD	EF	GH	I+	ab	cd	ef	gh	ij	kl	mn
op	qr	st	uv	wx	yz	@'	? ;	± !	$*	%⌑	$^{1\,2}$	$^{3\,4}$	$^{5\,6}$	$^{7\,8}$	$^{9\,0}$	$^{-\,+}$	$^{(\,)}$	°¬	( )	≠+	> <	≤ ≥	¢ =	[ ]	} {	└┘	┌┐	° _	—|

XN IMAGE

12	34	56	78	90	, .	AB	CD	EF	GH	IZ	JK	LM	NO	PQ	R*	$S	TU	VW	XY
12	34	56	78	90	, .	AB	CD	EF	GH	IZ	JK	LM	NO	PQ	R*	$S	TU	VW	XY
12	34	56	78	90	, .	AB	CD	EF	GH	IZ	JK	LM	NO	PQ	R*	$S	TU	VW	XY
12	34	56	78	90	, .	AB	CD	EF	GH	IZ	JK	LM	NO	PQ	R*	$S	TU	VW	XY
12	34	56	78	90	, .	AB	CD	EF	GH	IZ	JK	LM	NO	PQ	R*	$S	TU	VW	XY
12	34	56	78	90	, .	AB	CD	EF	GH	IZ	JK	LM	NO	PQ	R*	$S	TU	VW	XY

YN IMAGE

123	456	789	0ST	ABC	DEF	GHI	JKL	MNO	PQR	UVW	XYZ	*, .	
123	456	789	0ST	ABC	DEF	GHI	JKL	MNO	PQR	UVW	XYZ	*, .	#-$
123	456	789	0ST	ABC	DEF	GHI	JKL	MNO	PQR	UVW	XYZ	*, .	
123	456	789	0ST	ABC	DEF	GHI	JKL	MNO	PQR	UVW	XYZ	*, .	
123	456	789	0ST	ABC	DEF	GHI	JKL	MNO	PQR	UVW	XYZ	*, .	#-$
123	456	789	0ST	ABC	DEF	GHI	JKL	MNO	PQR	UVW	XYZ	*, .	

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.
- Hit LOAD.
- 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.
- Hit STORE.
- Turn off the ADDRESS COMPARE switch.
- Hit START.

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.
- Hit 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.
- Hit STORE.
- Turn off ADDRESS COMPARE.
- Hit START.

HOW TO BYPASS A CONSOLE MALFUNCTION

If a console malfunction occurs with any console device, the system normally sounds an alarm and suspends console activity. A malfunction can be recognized by the appropriate indicator lights on the console. One of two actions can be taken:

1. Correct the malfunction and make the console not ready, then ready. Processing will continue using the same console. The operation will be retried.
2. If an alternate console is available, hit the external interrupt key to switch to the alternate console. Next, make the alternate console not ready, then ready, and issue an operator command (for example, DISPLAY ACTIVE) on the alternate console. Processing will continue using the alternate console.

These procedures should be used for all console devices.

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 com-

mand 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 of the CCW) 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 THE SHARED DASD OPTION

The shared direct access storage device (DASD) option extends the operating system with PCP, MFT, or MVT by letting independently-running systems share such devices as the IBM 2311 and 2314.

The option is included in the operating system during system generation. The option allows two or more CPUs to efficiently use a shared device in a controlled way. This offers such advantages as:

- Lessening the amount of time you have to spend moving volumes from one system to another.
- Minimizing the updating of data sets -- you have to update only one instead of two or more duplicates.
- Simplifying scheduling -- you can run a job needing a specific data set on a shared device on any of the sharing systems, that is, you don't have to run it on one and only one system, unless the job has some other requirements, such as that it needs a unique set of devices, a certain control program (MFT for example), or extra amounts of main storage that all the systems don't have.

If your installation is using this option, you may receive special instructions from your management on what to do in certain cases. For example, you may be told:

- How you should keep track of the status of shared devices -- whether they are used by only one system for certain runs, whether you can change their status, on whose authorization, and how.
- How you should tell other operators in your installation when a parallel set of procedures must be carried out on all sharing systems. For example, when a VARY command is necessary on your system to change the status of a shared device, the operators on the sharing systems must also issue VARY commands to maintain a correct combination of mount characteristics.
- What you should do in the case of an enabled wait state on one or more CPU's - a general discussion of this topic is covered for each system under the topic "How to determine system status."

These and other considerations are wholly dependent on your installation's needs. You must get directions on what to do in specific instances from your installation supervision. The following instructions and guidelines are only to be used in conjunction with what your installation has already decided to be the best course of action.

HARDWARE NOTES

A two-channel hardware switch allows the control programming to actually carry out the reserving and releasing of the shared device and data. The two-channel switch allows a common control unit to be switched on a first-come, first-served basis between two channels, each from a different system.

On systems with the 2314, be sure any hardware toggle switches marked tagged and untagged are always set to the tagged position.

IPL FROM A SHARED DEVICE

An attempt to IPL from a shared device will hang if the control unit or device is busy with the other sharing system. Retry the IPL until successful. No adverse affects will arise in the other systems.

SYSTEM RESET

System reset only resets the functions of a shared control unit or device that belongs to the system that was reset -- any function having to do with another system is undisturbed. A selective or malfunction reset has no effect on device reservations or status.

MOUNT CHARACTERISTICS

Volume characteristics, device status, and volume mounting and demounting are all affected by the shared DASD option. One of the following combinations of mount characteristics and status must be in effect for each device which is physically shared between CPUs.

	System A	Other Systems
1.	Permanently Resident	Permanently Resident
2.	Reserved	Reserved
3.	Removable	Offline
4.	Offline	Reserved or Removable -- if a device is removable in one system, it must be offline in all others.

No other combination of mount characteristics is supported. You and your fellow operators must maintain the proper relationships between systems since the systems themselves do not and cannot.

Note: Device allocation must not proceed until valid mount characteristics are established. If the correct combination of mount characteristics is not maintained, the system may issue an unsatisfiable allocation request to which your only alternative is to cancel the job requiring the action; that is, the system may ask you to mount or demount a volume currently being used by another CPU.

Initializing Mount Characteristics

After initial program loading, a valid set of mount characteristics must be established before the system begins device allocation. This can be accomplished in one of the following two ways:

1. By specifying mount characteristics of shared devices in PRESRES, as described in IBM System/360 Operating System: System Programmers Guide, Form C28-6550.

2. By varying all shareable devices offline prior to issuing START commands and then following the parallel mount procedures (described in the next section) to establish appropriate mount characteristics. However, if you use this method, do not allow readers, writers, or initiators to start automatically, that is, device allocation must not begin until the valid mount characteristics are established.

Changing Mount Characteristics

The mount characteristics of a volume or the status of a device or both can be changed in one system as long as the resulting combination across all systems is valid.

Volumes that aren't permanently resident must be put in reserved status via the MOUNT command before jobs needing the shared volume are started.

To change volumes that are "reserved" you must:

- Use the VARY command to put the device in offline status in each sharing system and wait for the offline message in each system. If the system is in a quiesced state (no jobs in process), the offline message will not appear on the console.

- Use the MOUNT command to notify each sharing system of the units on which the new volume is being placed, and to put that volume in reserved status.

- Mount the volume.

- Use the VARY command to put the shared device in online status in each sharing system with PCP or MFT. In a system with MVT you will receive a normal allocation recovery message.

In MFT or MVT systems, no job requiring this new volume should be selected by an initiator before the volume is mounted. Selection before the volume is mounted could result in the abnormal ending of the job or could allow the volume to be allocated to a nonshared device when sharing is desired.

Job selection can be held up by:

1. Using the TYPRUN=HOLD parameter on the job card, or
2. By using the HOLD command, or
3. By assigning the job to a job class and not starting an initiator to that class. After the volume is mounted, use the RELEASE command to let the job be selected in cases one and two, start an initiator to the job class in the third case.

Any shared device can be allocated by only one system by using the VARY command in all the other systems to put the device in offline status in those systems.

OPERATING GUIDELINES

Just how complicated your job is when using the shared DASD option depends on what your installation has told you to do. You may or may not have to change the mount characteristics of a volume or device and perform parallel procedures on sharing systems. Thus the following statements are only general guidelines that don't necessarily apply to your installation.

MOUNTING A SHARED VOLUME

If shared devices are not permanently mounted, you must mount a new volume on a shared device before the job needing that volume is started. If a shared volume is not available when the requesting job is started, you must cancel the job. Don't reply to the allocation recovery message with the three-character device address of an offline shared device unless the device is offline on all sharing systems.

Don't change volumes until jobs using them are shown as ended by job ended messages displayed in response to your DISPLAY JOBNAMES command. When all of the jobs needing the volume currently mounted have completed, follow the procedures described under "Changing Mount Characteristics" above.

In systems with MFT or MVT, however, you must also temporarily suspend the selection of all jobs requiring volumes not yet mounted. Use the HOLD command, or the TYPRUN=HOLD parameter on the job card, for each job needing the volume. Later use the RELEASE command for each job, after the required volume has been mounted. Or you could assign the job to a job class but not start an initiator to that class until after the volume has been mounted.

Data Sets That Can Be Shared

Any of your installation's own data sets can be shared. In addition, the following system data sets can be shared:

- SYSCTLG when not on the IPL volume
- Volume tables of contents (VTOCs) on all shared volumes
- SYS1.LINKLIB if all sharing systems have identical operating system configurations
- SYS1.PROCLIB when not on the IPL volume.

Data Sets That Cannot be Shared

The following system data sets cannot be shared.

- SYSCTLG when on the IPL volume
- SYS1.SVCLIB
- SYS1.NUCLEUS
- SYS1.LOGREC
- SYS1.SYSJOBQE
- SYS1.SYSLOGX
- SYS1.SYSLOGY
- SYS1.ROLLOUT
- SYS1.ACCT
- PASSWORD
- SYS1.ASRLIB

TAKING POWER DOWN

When taking power down on any system using the shared DASD option, be sure to follow this sequence:

- Hit RESET
- Disable the shared control units connected to the system
- Hit POWER OFF

SUMMARY

Be sure that you know and follow all your installation's rules and procedures when you operate a system with the shared DASD option. Remember that sharing systems cannot communicate with each other and that therefore it is up to you and your fellow operators to cooperate fully to avoid errors.

You can resume system activity after entering an enabled wait state on all CPUs by using the CANCEL command to cancel one or more jobs as necessary.

When you perform initial program loading on a system, you must ensure that the proper mount characteristics are established before allowing device allocation to proceed. You must know which devices are shared and must not reply to an allocation recovery message with a shared device address unless you are in the process of mounting to that device in an MVT system.

Whenever you change the mount characteristics of a shared volume or the status of a device, or both, you must maintain a valid combination of these characteristics across all sharing systems. Remember that the VARY OFFLINE command is always the first step of the mounting procedure and that the procedure must be done in parallel on all sharing systems.

If there is a hardware malfunction on other than the system residence device, you must vary the device offline on all sharing systems. You can then mount the shared volume on another shared device, if one is available, as long as you follow parallel mount procedures on all sharing systems.

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 (IBCRCVRP), 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 hitting 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, hit 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 don't 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.

  Hit 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

The operating system recovery management programs record hardware bugs in the CPU and channels in Models 40, 50, 65, and 75, and in some cases attempt to recover from these failures.

These optional recovery management programs include the system environment recording (SER0 or SER1), machine check handler (MCH), and channel check handler (CCH) routines. The MCH and CCH routines are only on Model 65.

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 recovery management programs, 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 a recovery management program. 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. 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.

If MCH receives control, it analyzes the error and attempts a recovery by retrying the failing instruction if possible. If retry is not possible or is unsuccessful, MCH will attempt to repair the malfunction, or isolate it to a task, or both. If the program damage cannot be repaired, the affected job is ended abnormally, and the system operation can continue. If neither retry nor ending a single job is possible, the system is put in the wait state. In any case, MCH formulates an error record, writes it out on the SYS1.LOGREC data set, and informs you of results in a message.

If a channel check occurs and if CCH gets control, CCH will attempt error recovery on channel errors and these errors will be recorded on the SYS1.LOGREC data set.

When the SYS1.LOGREC data set has been filled -- by any of the recovery management routines -- print out the data set's contents by using the environment recording edit and print routine, called IFCEREP0. This routine formats and writes the records on SYS1.LOGREC onto printer, tape, or disk according to your installation's specifications.

IFCEREP0 is described in the publication IBM System/360 Operating System: Utilities.

To use IFCEREP0 to write out the contents of the SYS1.LOGREC data set and at the same time clear the data set for reuse, you can use a JOB card followed by:

```
//         EXEC   PGM=IFCEREP0
//SERLOG   DD     DSNAME=SYS1.LOGREC,        x
//                DISP=(OLD,KEEP)
//EREPPT   DD     SYSOUT=A
```

HOW TC 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.
- Hit 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 VCLUME LABELS

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

CONTRCL 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.

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 and MFT. The delimiter statement format is:

```
┌--------------------┐
|/*                  |
└--------------------┘
```

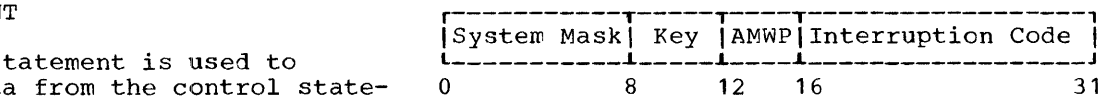
THE CURRENT PSW CODE

When the machine goes into a WAIT state, certain codes can be displayed in the low-order 12 bits of the current PSW to give an indication of what happened.

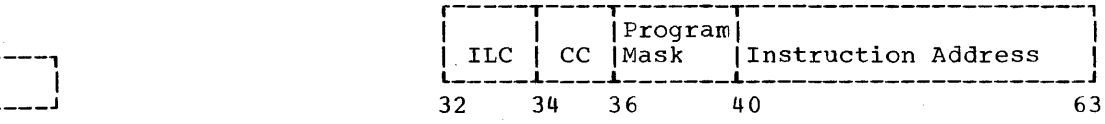
These codes, their explanations, and your responses are given in the publication IBM System/360 Operating System: Messages and Codes.

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



Normal Right Half of Current PSW



MODEL 30 TECHNIQUES

DISPLAYING THE CURRENT PSW

Storage displaying, including displaying the current PSW, is done one byte at a time on the Model 30 (Figure 16).

- Hit 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.

- Hit 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.

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.

DISPLAYING MAIN STORAGE

- Hit 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.
- Hit DISPLAY. The data will be in the main storage data register, while the address will be in the main storage address register.

CLEARING MAIN STORAGE

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

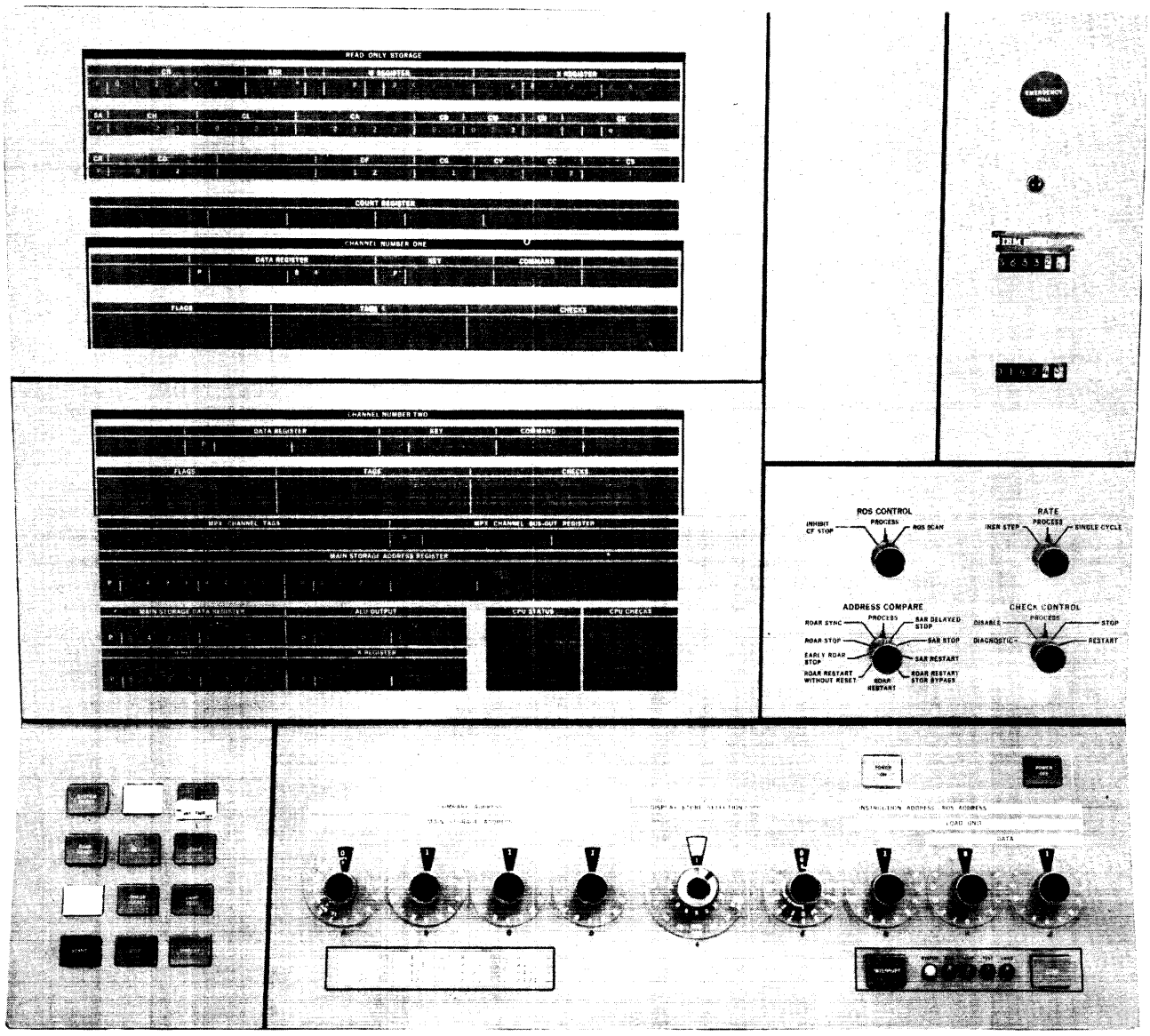


Figure 16. Model 30 System Control Panel

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:

FPR byte	Switch D setting
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.
- Hit 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.

MODEL 40 TECHNIQUES

DISPLAYING THE CURRENT PSW

Storage displaying, including displaying the current PSW, is done two bytes at a time on the Model 40 (Figure 17).

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

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

- Hit DISPLAY.

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.

DISPLAYING MAIN STORAGE

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

CLEARING MAIN STORAGE

- Hit STOP.
- Disable the interval timer.
- Hit SYSTEM RESET.
- Set the RATE switch to SINGLE CYCLE.
- Enter 1000 (hexadecimal) in STORAGE DATA keys.
- Flip up the STORE STATS switch.
- Set RATE switch to PROCESS.
- Set DIAGNOSTIC CONTROL switch to MS ADDRESS.
- Hit 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.
- Hit SYSTEM RESET.

ALTERING MAIN STORAGE:

- Set up address as you would in displaying main storage.
- Set the halfword of data to be entered into the data keys.
- Hit 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 halfword desired in the portion marked HALFWORD.
- Hit DISPLAY. The halfword 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 halfword desired in the HALFWORD portion.
- Hit DISPLAY. The halfword will be displayed in the storage data lights.

ALTERING REGISTERS

- Follow same procedure as in displaying a register.
- Put data in storage data keys.
- Hit STORE.

STOPPING ON ADDRESS COMPARE

- Put the address you wish to stop on in the storage address keys.
- Turn ADDRESS COMPARE to MS STOP.

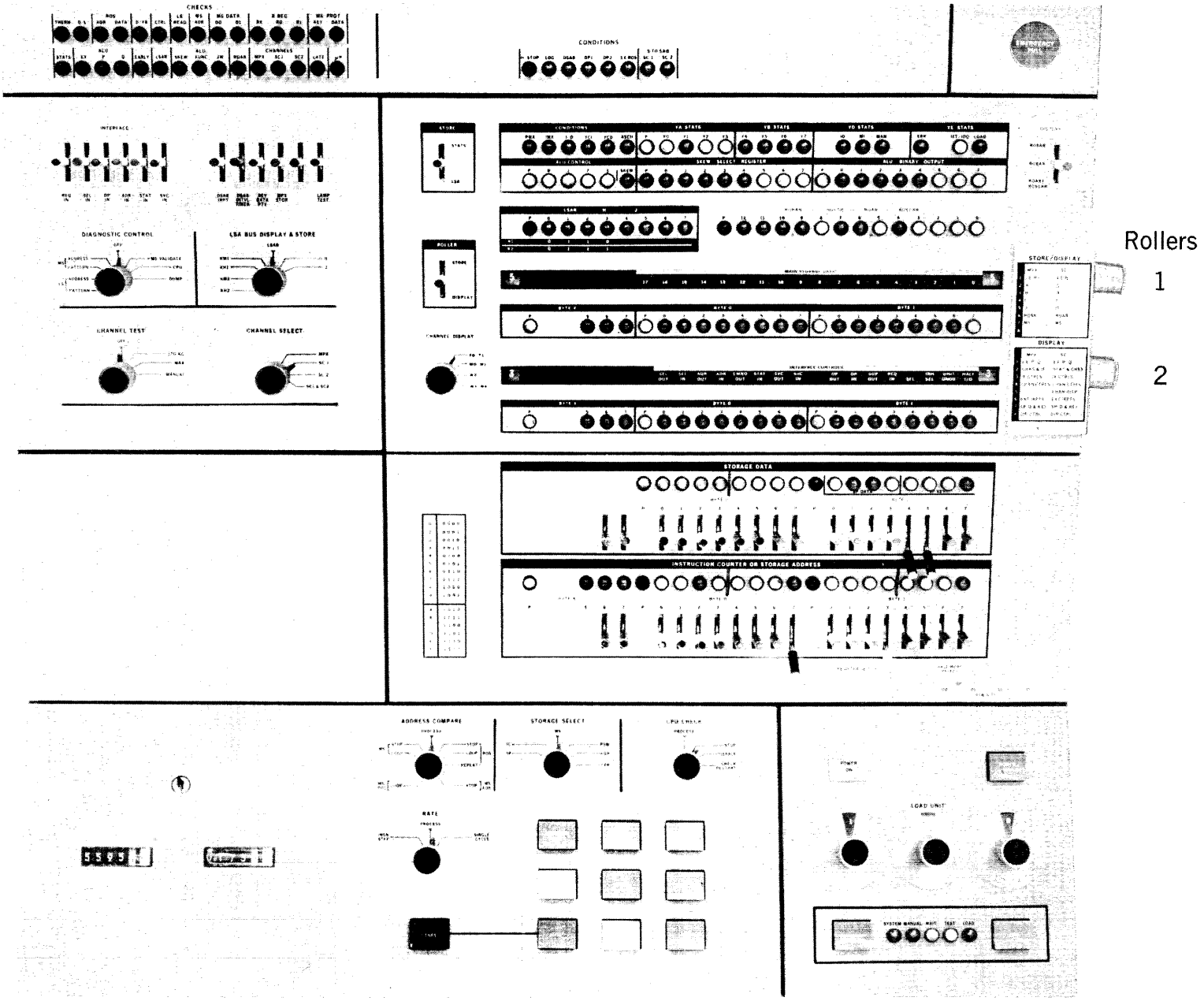


Figure 17. Model 40 System Control Panel

MODEL 50 TECHNIQUES

DISPLAYING THE CURRENT PSW

Storage displaying, including displaying the current PSW, is done 4 bytes (one word) at a time on the Model 50 (Figure 18).

- Hit STOP.
- Set the ADDRESS switches to 170.
- Set the STORAGE SELECT switch to LOCAL.
- Hit DISPLAY.
- 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.

ALTERING THE CURRENT PSW

- Be sure machine is in manual mode.
- Set the ADDRESS switches to 170.
- Set the STORAGE SELECT switch to LOCAL.
- Put new data in the DATA switches.
- Hit STORE and the new data will be entered.

DISPLAYING MAIN STORAGE

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

CLEARING MAIN STORAGE

- Hit STOP.
- Hit SYSTEM RESET.
- Set IC to zero.
- Set RATE switch to SINGLE CYCLE.
- Put 0200 in the DATA switches.
- Flip ROS REPEAT INSN down.
- Hit START.
- Set RATE switch to PROCESS.
- Flip up all DATA switches.
- 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 switches.
- Set the STORAGE SELECT switch to MAIN.
- Put data to be stored in the DATA switches.
- Hit 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 switches 24-27.
- Set STORAGE SELECT switch to LOCAL.
- Hit DISPLAY. 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 switches 24-27.
- Set STORAGE SELECT to LOCAL.
- Put data in the DATA switches.
- Hit 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 switches.
- 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.

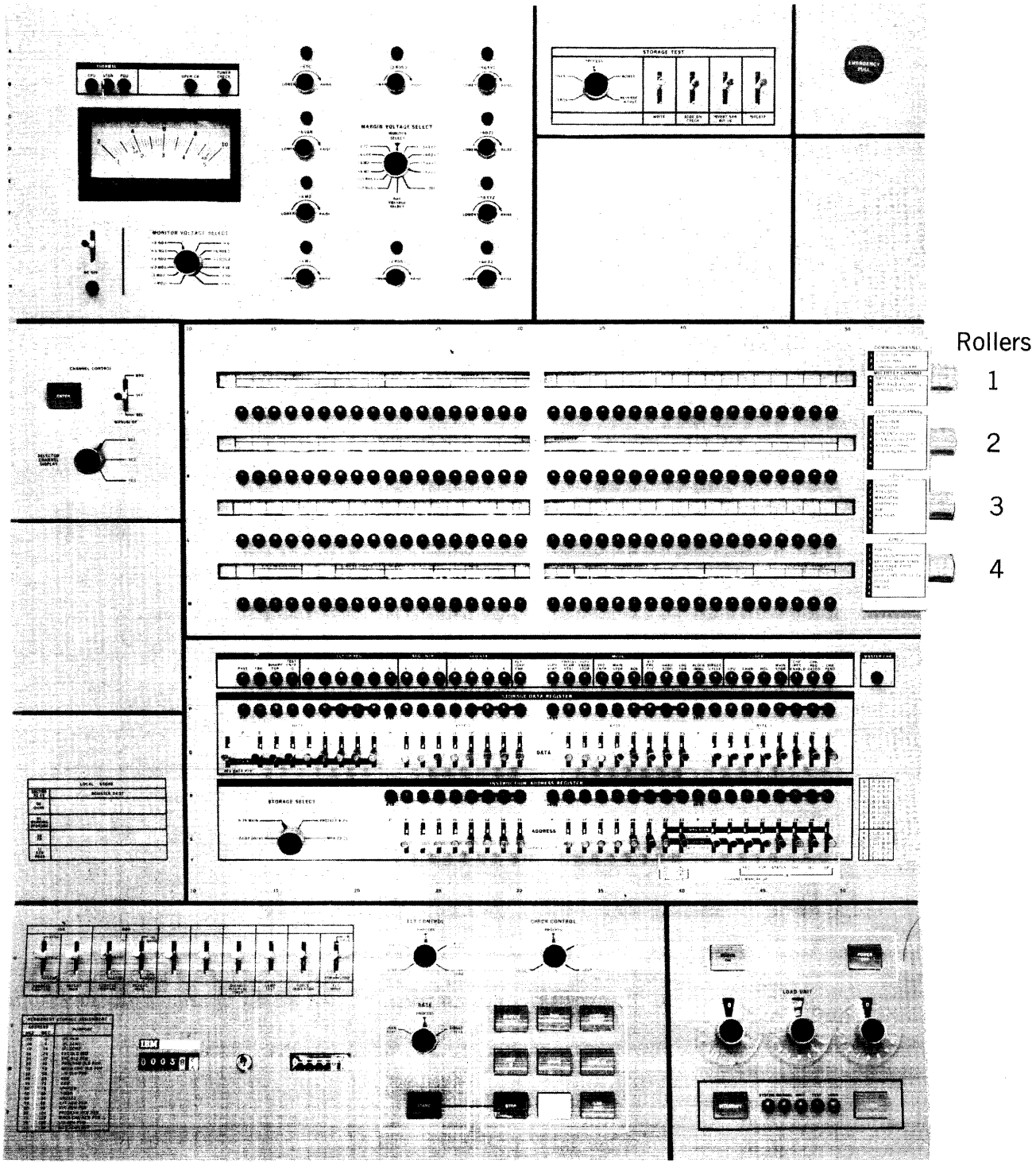


Figure 18. Model 50 System Control Panel

MODEL 65 TECHNIQUES

DISPLAYING THE CURRENT PSW

Storage displaying, including displaying the current PSW, is done eight bytes (two separate words) at a time on the Model 65 (Figure 19).

- Hit 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 instruction address.

ALTERING THE CURRENT PSW

- Put new PSW doubleword in the data keys.
- Store the data at location zero.
- Hit PSW RESTART.

DISPLAYING MAIN STORAGE

- Hit STOP.
- Set the STORAGE SELECT lever switch to the middle position -- main storage.
- Set the address switches to the address to be displayed.
- Hit 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).

CLEARING MAIN STORAGE

- Hit 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.
- Hit 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.
- Hit 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.
- Hit START.

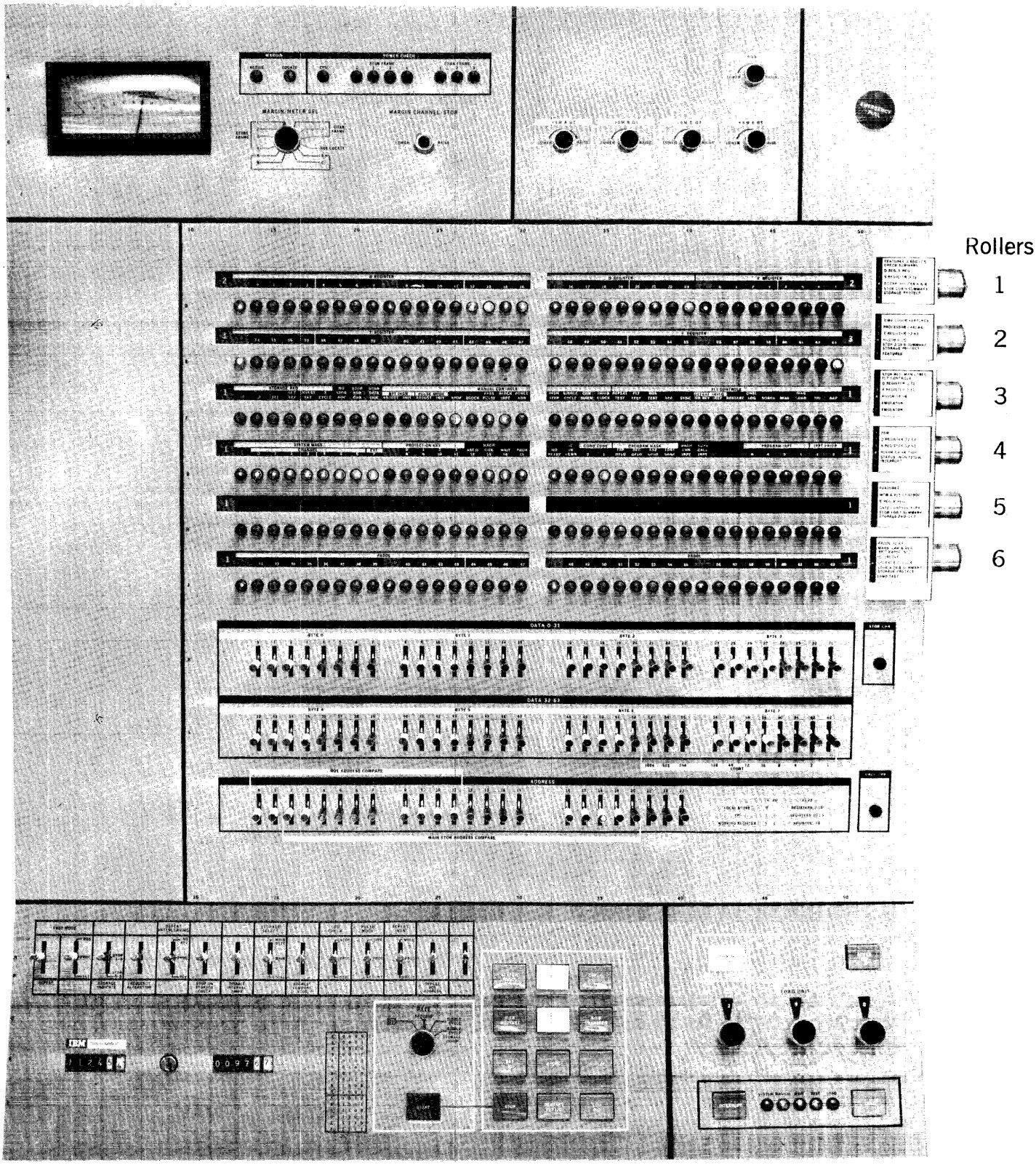


Figure 19. Model 65 System Control Panel

MODEL 75 TECHNIQUES

DISPLAYING THE CURRENT PSW

Storage displaying, including displaying the current PSW, is done eight bytes (one whole doubleword) at a time on the Model 75 (Figure 20).

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

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.

DISPLAYING MAIN STORAGE

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

CLEARING MAIN STORAGE

- Hit STOP.
- Hit 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.
- Hit LOAD A-B REGS.
- Hit START.
- Flip up ENABLE STORAGE RIPPLE.
- Hit SYSTEM RESET.

ALTERING MAIN STORAGE

- Hit STOP.
- Set address switches to the location of the lowest byte of the doubleword to be altered.
- Set STORAGE SELECT to MAIN STOR.
- Hit DISPLAY. Contents of doubleword will be displayed in J register.
- Set new information in the appropriate data switches.
- Hit 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.
- Hit START.

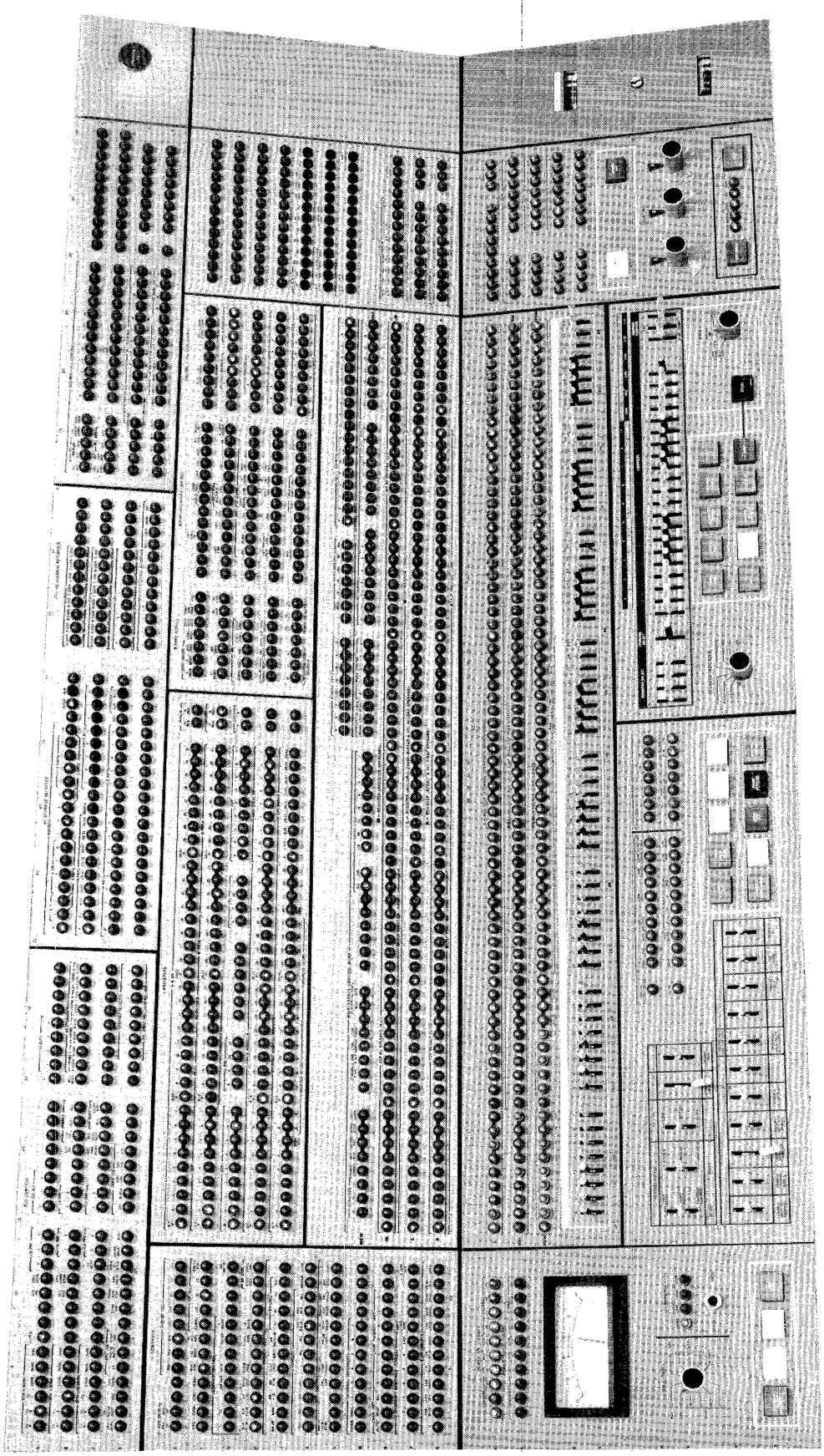


Figure 20. Model 75 System Control Panel

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

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 heading "Displaying 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

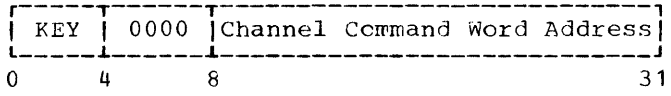
Input/Output Old PSW -- Hex location 38

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

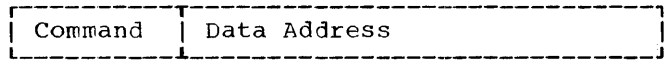
The CAW specifies the storage protection key and the address of the first channel command word associated with the START I/O instruction. The CAW is found at hex location 48.

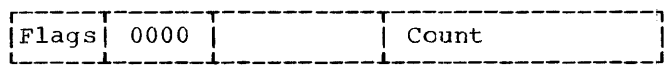


CHANNEL COMMAND WORD

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 any-

where in main storage, and can exist singly or in a group called a channel program.





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).

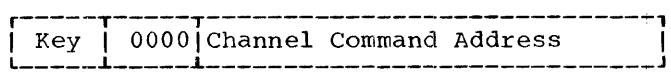
Bits 37-39 contain zeros.

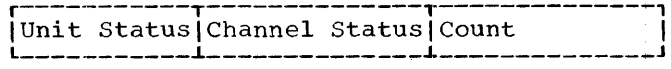
Bits 40-47 are ignored.

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.

OPERATING HINTS

How to Handle Punched Cards

- Avoid nicking of cards -- hold a card deck by its front and back rather than by its edges.
- When removing cards from a stacker, ruffle them before you place them into a card tray; be sure to tightly clamp the card compressor in the tray.

How to Handle Tape Reels

- Avoid dust or any other foreign substances on tape; touch a tape as little as possible; keep tapes in tape cases or in tape seals when the tapes are not in use.
- Hold a tape reel always at its hub rather than by its rims.
- Write on or post tape stickers (if required) before you paste them on tape reels.
- Keep tape drive windows closed, except when mounting or demounting tape reels.
- When mounting a tape reel, keep the tape taut and be sure the load point

has well passed the read/write heads before you close the tape drive and hit LOAD REWIND.

How to Handle Disk Packs

- Avoid dust or any other foreign substances on the disk surfaces.
- To label disk packs, use installation approved stickers on the recessed portion of the hub or write your disk pack identification on the hub using a pen; do not use a lead pencil.
- Never use an abrasive such as an eraser.
- If stickers are used to label disk packs, write on or post stickers before you paste them on disk packs.
- To store disk packs, be sure you (1) place them flat on a shelf, (2) place one disk pack beside the other (never one on another), and (3) keep them out of direct sunlight.
- To mount a disk pack, (1) set the pack on the drive spindle, (2) turn the disk pack top cover (by its handle) clockwise until it stops, (3) lift the cover from the pack, and (4) close the cover of the drive.
- To demount a disk pack, (1) turn the disk pack top cover counterclockwise until it locks in (at least two full turns) and (2) lift the pack off (vertically until clear of drive) and firmly fasten the bottom cover to the disk pack. Close the disk drive if you don't mount another disk pack.

How to Handle Data Cells

- Keep the two-piece protective cover closed at all times, except when you must open it to install the data cell or to return the data cell after this has been removed from the data cell drive.
- To store data cells, be sure you (1) keep them out of direct sunlight and (2) place them in your storage location one beside the other (never one on another).
- To install or remove a data cell, be sure to check the operator panel for

  AC = on,
  Drive Op = on,
  Drive Ready = on,
  DC = on, and
  Drive Select = off

  before you open the entry door.
- To label a data cell, use a sticker (preferably IBM P/N 2174086) on the front surface of the cell. Use a pen (never a lead pencil to write on, post, or otherwise alter a data cell label. You must never use an abrasive such as an eraser.

1403 Printers

- To insert new forms, you can, after having opened the print gate and the tractor covers, lift up the partially printed page and underlay the new forms.
- To install a carriage control tape, be sure the numbers on the tape are on the outside and the locking knob is adjusted for no noticeable slack.
- When a paper runaway occurs, hit CARRIAGE STOP on the printer and check the carriage control tape.

  If the tape is worn, replace it by a new one.

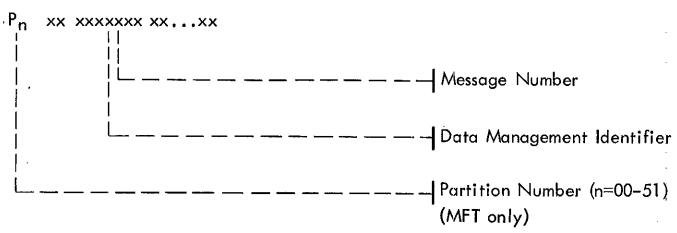
  If the tape is in good condition, cancel the job via your console and report the reason for the cancelation to your programmer.

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 and Codes.

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

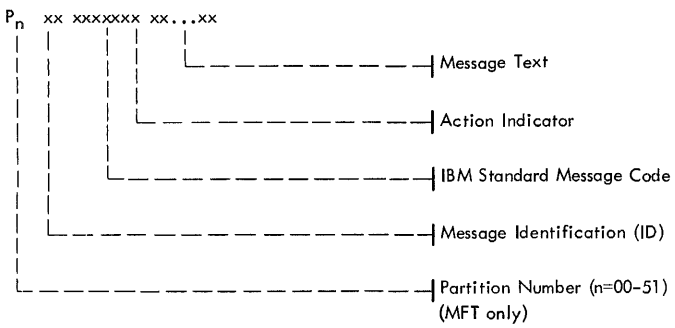
The IBM standard message code is a unique alphameric code of at least six characters assigned to each message written by IBM programs. The code identifies the program that is the source of the message.

For example, codes starting with the following characters identify the programs listed:

Code	Program
IBC	Independent Utilities
IEA	Supervisor and NIP
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)
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	Linkage Editor
IEX	ALGOL Compiler
IEY	FORTRAN (G)
IFB	System Environment Recording: SER0, SER1, OBR, SDR
IFC	Environment Recording Edit and Print
IFD	Online Test Executive
IGF	Machine Check Handler (MCH)
IHB	Supervisor and Data Management
IHC	Library Subroutines (FORTRAN)
IHE	Library Subroutines (PL/I)
IHG	Update Analysis
IHI	Library Subroutines (ALGOL)
IHJ	Sort Checkpoint/Restart
IHK	Remote Job Entry (RJE)
IKA	2250 Job Control

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 one-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 5.

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 demount unlabeled tape volumes.

If you're 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 demounted.

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.

background job: A job that is processed in a partition or region other than the one associated with the input/output unit that was used to define this job.

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.

cursor: See cursor symbol.

cursor symbol: A short line (underscore) displayed on the 2250 screen to indicate where the next character typed on the 2250 keyboard will be positioned.

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.

foreground job: A job that is processed in a partition or region associated with the input/output unit from which the job was defined, thus preventing the input/output unit from being used to define another job until processing of the current job has been completed.

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.

GJP: (See graphic job processor.)

graphic job processor (GJP): A program that elicits job control information from a user as he selects and performs job control operations at an IBM 2250 Display Unit. It interprets the information entered by the 2250 user and converts it into job control language.

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 collective term for the 15 queues of job information which the job scheduler uses to select the jobs and job steps to be processed. Each of the 15 queues is associated with one input job class.

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.

light pen: A pen-shaped instrument that can be used to sense light at a point on the 2250 screen.

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-adjoining 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 that contains the IPL program, the volume index of the SYSCTLG data set, and the system data sets SYS1.NUCLEUS, SYS1.SVCLIB, and SYS1.LOGREC. The system residence volume must reside on the I/O device which is addressed when initial program loading is performed.

SYSCTLG: A 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 neither permanently resident in main storage nor 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 system 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, machine check handler recovery management 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.

time slicing: The revolving assignment of CPU time. The supervisor limits the amount of processing time that each task in a time-sliced group receives at any one time. After the tasks have all had their slice, the sequence is repeated again and again. Not all tasks the system is performing need be time-sliced. If some are not, they are given control according to priority, competing with the time-sliced group as a whole for CPU time.

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.

READER'S COMMENT FORM

IBM System/360 Operating System
Operator's Guide

Form C28-6540-7

- Is the material:

	Yes	No
Easy to read?	☐	☐
Well organized?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Accurate?	☐	☐
Suitable for its intended audience?	☐	☐

- How did you use this publication?
  - ☐ As an introduction to the subject
  - ☐ For additional knowledge

  Other .....

- Please check the items that describe your position:
  - ☐ Customer personnel
  - ☐ IBM personnel
  - ☐ Manager
  - ☐ Systems Analyst
  - ☐ Operator
  - ☐ Programmer
  - ☐ Customer Engineer
  - ☐ Instructor
  - ☐ Sales Representative
  - ☐ Systems Engineer
  - ☐ Trainee
  - Other .....

- Please check specific criticism(s), give page number(s), and explain below:
  - ☐ Clarification on page(s) .....
  - ☐ Addition on page(s) .....
  - ☐ Deletion on page(s) .....
  - ☐ Error on page(s) .....

Explanation:

- 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

IBM®

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

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

Printed in U.S.A. C28-6540-7

C28-6540-7

Printed in U.S.A. C28-6540-7

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]