Z77-7262

TECHNICAL INFORMATION EXCHANGE

IBM®

October 12, 1967

SYSTEM '360 OPERATING SYSTEM

OPERATOR TRAINING COURSE

T. A. Clarke
IBM Corporation
570 Broad Street
Newark, New Jersey 07102

This S/360 OS Console Operator Training Course is intended for use immediately after the operator has completed the OS Operator Training Programmed Instruction Course, including the S/360 machine exercise. Its purpose is to explain the general functional concept of OS, its various components and their respective roles, to the extent that the console operator has a good knowledge of what is happening under OS processing. Topics covered in the OS Operator PI Course are elaborated upon; other topics not covered in the PI are discussed to some extent. This paper discusses primarily OS Primary Control Program and is current with Release 11 of OS. Several OS MFT concepts are discussed. Hard copies of foils used are included in the package.

For IBM Internal Use Only

IBM Corporation, Technical Publications Dept., 112 E. Post Road, White Plains, N.Y. 10601

Z77-7262

* CONTENTS *
(Course VTOC)

SUBJECT	SECTION	FOIL
Table of Contents	1-2-3	
Course Objective	4	
Course Sequence	4	
Course Time Estimates	4	
Course Reference Manuals	5	
Definition of Operating System	I A	
Reasons for an Operating System	I B	1
Objectives of the Operating System	I C	1
O/S Vocabulary Definitions Index	II A	
Volume	II B	8
Data Set	II C	9
Job Control Language	II D	10
Job and Job Steps	II E	11
SYSGEN	II F	13
SYSRES	II G	13
Volume Lable-Direct Access	II H	14
VTOC	II I	14-15-15.1
Volume Lable - Tape	II J	16-16.1
Additional Terms	II K	
O/S General Organization/Function	III A	2
O/S DATA MANAGEMENT RTNS	III B	3
O/S JOB MANAGEMENT RTNS	III C	4
O/S TASK MANAGEMENT RTNS	III D	5
O/S Processing Programs	III E	6
O/S Conceptual Summary	III F	7
O/S Components on DASD	IV A	18
O/S Library/Catalog Facility	IV B	19
SYS1.NUCLEUS	IV C.1	20-21
SYS1.LOGREC	IV C.2	20-21
SYS1.SVCLIB	IV C.3	22-23
SYSCTLG	IV C.4	22-23
SYS1.LINKLIB	IV C.5	24-25
SYS1.SYSJOBQE	IV C.6	24-25
SYS1.PROCLIB	IV C.7	26-27
SYS1.MACLIB	IV C.8	28-29-30
SYS1.MODLIB	IV C.9	28-29-30
SYS1.GENLIB	IV C.9	28-29-30
O/S Components on SYSRES	IV D	32
General Concept: Core-Res O/S	V A	33
Main Storage Map	V B	34
Scheduler/Supervisor Consideration	V C	36-35
O/S Scheduler	V D	36-36.1-36.2
O/S Supervisor	V E	35
O/S - Operator Communication	VI A	37
O/S 1052 Message Format	VI B.1	38

SUBJECT	SECTION	FOIL
Message ID	VI B.2	39
Standard Code	VI B.3	40-41
Action Indicator	VI B.4	42
Message Text	VI B.5	42
Problem Program 1052 Messages	VI C	43
1403 Messages from O/S	VI D	44
Displayable PSW Codes	VI E	45
Console Alarm Bell	VI F	
Operator Commands (PCP)	VI G.1	46
CANCEL	VI G.2	47
DISPLAY	VI G.3	48
MOUNT	VI G.4	49
REPLY	VI G.5	50
REQ	VI G.6	51
SET	VI G.7	52
START	VI G.8	53
STOP	VI G.9	54
UNLOAD	VI G.10	55
VARY	VI G.11	56
Command via System Reader	VI H	57
General Operation of O/S - PCP	VII A	34
Detailed Operation of O/S - PCP	VII B	
IPL	VII B.1	58-59
NIP	VII B.2	59-60
Load Error	VII B.3	60-61-62
SET Command	VII B.4	63
START Command	VII B.5	64
VARY Commands	VII B.6	65-66
Job Control Language	VII B.8.1-5	67 to 74
Detailed Operation: JCL	VII B.9	75-76
VTOC Search	VII B.10	75-76
Program Fetch	VII B.11	77-78
Execute/Terminate	VII B.12	79 50 81

Course Objective: This course is intended for use immediately after the system console operator has completed the O/S Operator Training Program Instruction course, including the S/360 O/S machine exercise, referred to below. It's purpose is to explain the general functional concept of O/S, it's various components and their respective roles to the extent that the operator has a good knowledge of what is happening under O/S processing. Topics covered in the PI Course are elaborated upon; other topics not covered in the PI are discussed.

Course Sequence: It is intended that this course be preceded by the following courses or their equivalent:

System/360 Introduction
S/360 Model 30 Console Operations - Native 360 Mode & 1401 Compatibility Mode
S/360 Advanced Operator Concepts
S/360 O/S Operator Training PI

Course Time Estimates: Allowing for questions from 10 students and a moderately detailed explanation of JCL (Section VII B.8), this course should take approximately 16 hours.

IBM Manuals referred to in preparation of this course are listed below:

Title	Form
S/360 Principles of Operation	A22-6821
Operating System/360 Introduction	C28-6534
Operating System/360 Option 2: MFT Concepts and Facilities	C27-6926
Operating System/360 Concepts and Facilities	C28-6535
Operating System Data Management	C28-6537
Operating System Job Control Lang.	C28-6539
Operating System JCL Charts	C28-6632
Operating System: System Generation	C28-6554
Operating System Control Program Services	C28-6541
Operating System Operator's Guide	C28-6540
Operating System Messages, Completion Codes and Storage Dumps	C28-6631
O/S Introduction to Control Program Logic-PLM	Y28-6605
O/S Control Program with Option 2-PLM	Y27-7128
O/S Fixed Task Supervisor-PLM	Y28-6612
O/S Job Management-PLM	Y28-6613
O/S Catalog Management-PLM	Y28-6606
O/S I/O Supervisor-PLM	Y28-6616
IBM Systems Journal Volume 5, Number 1, 1966	

I A What is the Operating System on System/360?

Operating System/360 consists of a comprehensive set of language translators and service programs which operate under the supervisory control and coordination of an integrated set of control routines.

I B Why an Operating System?

Cope with problems of:

(1) Set up time for both physical mounting of devices and job-to-job transition,

(2) allocation of resources (CPU core and I/O devices) so as to keep the System as busy as possible by processing more than one program at a time in multi-program environment,

(3) associating priorities with the individual programs and,

(4) the need for symbolic input/output assignment.

Foil #1

I C.1 Objectives of an Operating System

A. Increase the THRUPUT of the System (that is, the total volume of work performed by the System over a given period of time.

Done by:

(1) Processing a continuous stream of job without interruption (job-to-job transition).

(2) Setting up (mounting packs and tapes) ahead of time.

(3) More efficient use of the physical and programming resources through multi-programming, accessing I/O devices via various channels, etc.

Foil #1

I C.2

B. Decrease the response (turn-around) time of the System per job (that is, reduce the time required to process a job from when initially submitted for processing until the time when the job is completed and response is received from the System;

Done by:

(1) Processing jobs on a single system from beginning to end, as opposed to the use of a large System, then one or more peripheral systems.

(2) Remote job entry made possible through the use of telecommunications.

Foil #1

C. To assist the manager of the installation.

Done by:

(1) In programming, by the modular approach to programming an entire run.

(2) Allowing priority codes to be assigned to jobs so that processing preferences (priorities) are established.

(3) Job accounting done with job processing.

(4) Aiding in keeping an accurate inventory of internally stored programs and data.

Foil #1

D. To assist the programming of problem solutions or applications on the system.

Done by:

(1) Broader and more powerful language translator facilities.

(2) Ability to link to precoded routines and programs.

(3) Provides the programmer with a system of controls regarding the status of processing or computer resources.

Foil #1

I C.5

E. To assist the operator of a system

Done by:

(1) Specific operator instructions that enable the operator to have a clear communication path with the system.

(2) Providing automatic systems logging.

(3) Releases the operator from data handling problems by keeping track of programs and sets of data.

Foil #1

II A O/S Vocabulary Definitions

Index of Definitions:

Page II B. Volume

C. Data Set

D. JCL

E. Job & Job Steps

F. SYSGEN

G. SYSRES

H. Volume Label on Direct Access (DASD)

I. Volume Table of Contents (VTOC)

J. Volume Label on Magnetic Tape

K. Additional Terms

II B Volume

A standard unit of auxiliary storage is called a volume. Each disk pack, data cell, drum or disk area served by one access mechanism (e.g. 2302) is considered a volume. Each reel of magnetic tape is considered a volume, each disk pack is considered a volume.

Foil #8

II C DATA SET (DS)

A data set is a named, organized collection of one or more records that are logically related. Under concepts of IBM Systems prior to System/360 a data set would have been termed "a file". The information in the data set is not restricted to any particular type of data, that is, a given data set might contain, as it's data, the actual tape records reflecting all employees on the payroll (the Payroll Master Data Set); or it might be a COBOL Source deck of cards used as an input data set to the COBOL compiler; or it might be an object deck (e.g. output from a translator program) which, instead of being in card form, is written on a 2311 disk as an object data set. The data set may reside on, be input from, or output to any type of I/O device.

Normally each different data set is assigned a unique data-set-name by the user; this name may be either a simple name or a qualified name. A simple name given to a data set must be composed of from 1 to 8 alphameric characters, the first of which must be alphabetic (e.g., PAYROLL1, DEBITPOL, X0003, ABC); a qualified data set name may be used, which would take the form of simple data set names separated by periods (e.g., DEBITPOL.ADDITION.MAY67,ABC.DEF.GHIJ.K0004, PAYROLL.NEWYORK).

Under S/360 O/S they may be differently organized data sets, and, depending on this organization, a specific programmed method of accessing the data in the data set would be used; the organizations and access methods available under O/S are described briefly on page II C.1.

Foil #9

II C.1 DATA SET ORGANIZATIONS & ACCESS METHODS

(SAM) Sequential Access Method - The physical records are organized in sequence; normally the records are accessed from the first to the last in sequence (e.g., tape, punch card, paper tape, printer).

(DAM) Direct Access Method - The records are organized in any manner; the data set must be located on a direct access device and records are accessed directly.

(ISAM) Indexed Sequential Access Method - Records are organized in logical sequence on tracks of a direct access device and additionaly, a separate index is maintained to point to certain principal records of the data set; records are normally accessed by reference to the index, then to the specific DASD track; records may also be processed sequentially.

(TAM) Telecommunications Access Method - This organization is given to data coming from or going to remote on-line teleprocessing terminals.

(PAM) Partitioned Access Methods - In concept, like an index sequential/ sequential organization; independent records (called members) are organized in the DS on DASD; a directory to the starting point of each member is maintained as part of the data set. Members are accessed by reference first to the directory, then to the member which is read sequentially from DASD.

These access methods may be used at a basic level or queued level by the programmer, thus the terms:

BSAM	QSAM
BDAM	
BISAM	QISAM
BTAM	QTAM
BPAM	

II D Job Control Language (JCL)

Job Scheduling within the S/360 Operating System is controlled by information placed in the input stream by the programmer. This controlling information is done by using a coding technique somewhat similar to a programming language; thus coding language is known as O/S Job Control Language (JCL). The JCL coding is presented to O/S in the form of job control statements which enter the system in the input stream (from the input reader I/O device) and may be in the form, for example, of cards or tape records. There are six types of JCL statements: Job, Execute, Data Definition, Command, Delimiter, and Null -- each of which is discussed somewhat more in Section VII of this course.

Foil #10

II E Job and Job Steps

A "job" is a grouping of one or more actual programs to be executed which are recognized to be related by the single JOB statement in the JOB CONTROL LANGUAGE statements read from the system input reader device. The program or programs which make up a job are known as job steps. A step of a job is recognized by the system by the EXEC (execute) statement in the JOB CONTROL LANGUAGE statements. The relationship of steps to a job can be seen, for example, with:

Job = Fortran

Step 1	Compile
Step 2	Link Edit
Step 3	Execute

Step 1 will bring in the Fortran Source Cards and compile them; the output of Step 1 is the object deck. Step 1 can pass this data set (the object deck) to Step 2 which will take the object deck as Step 2 input. Output of Step 2 (LINK EDIT) is a load module, which will be passed to Step 3. Step 3 (Execute) will take the load module as input, load it into core, give control to it and begin execution.

Foil #11

II F System Generation - (Sys Gen)

Sys Gen is a common abbreviation for the term system generation, which means the building, or the assembling of the operating system itself (just a lot of routines and programs) when it was received from IBM. This generation consists largely of describing the hardware configuration of your System/360 and specifying those special options desired by your installation. Numerous assemblies are performed, which result in the building of the routines which make your Operating System.

Foil #13

II G System Residence (SYS RES)

System Residence (SYS RES) is the name commonly given to the direct access storage device(s) (DASD) or disk pack(s) which houses the Operating System program. Most users of O/S now have their entire operating system housed on a single disk pack. This pack is known as SYS RES, which is where the system resides.

Foil #13

II H Volume Label on Direct Access

Each direct-access volume is identified by a volume label, which always contains the volume serial number of this volume and the location of the volume table of contents (VTOC).

The volume label is always in a standard location, that being Cylinder zero, track zero of the volume (actually record 3).

Other information contained in the DASD volume label is:

Vol1	4
Volume Serial Number	6
Vol Security	1
VTOC Pointer	10
Reserved	20
Owner Name and Address	10
Blank	29
	80 Bytes

Foil #14

II I Volume Table of Contents (VTOC)

Each direct access volume has a volume table of contents (commonly referred to as simply VTOC). This is a special data area, rather than a "library"; the VTOC contains entries showing the names of all data sets and the exact space (from cylinder xxx track xx to cylinder yyy track yy) allocated to each one of these data sets. These entries for each data set are known as Data Set Control Blocks (referred to simply as DSCB). When a specific data set is referrenced (for example: during the initiation of a job step, at OPEN time, at CLOSE time, and at END OF VOLUME), the VTOC has it's DSCB's searched to find the cylinder and track location of the desired data set.

The VTOC is initially written on a direct access device by the DASDI program, which is a stand-alone utility program to initialize such devices.

The VTOC may be printed out by using the IEHLIST program, which is an O/S utility program.

Foil #14 - #15 - #15.1

II J Volume Label on Magnetic Tape

Each reel of magnetic tape is considered a volume. In view of the serial properties of tape, the method used for identifying volumes and data sets on tape differs somewhat from the method used for direct-access devices. Volume labels and data set labels are used, however, each data set label exists in two parts -- a header label preceeding the data set and a trailer label following it. O/S uses a generalized labeling procedure which permits users to employ their own tape-labeling conventions. Unlabeled tapes may be used under O/S, but the responsibility for mounting the right tape becomes the operator's.

Vol 1	4	
Vol Serial Number	6	
Vol Security	1	
Data Set Directory	10	Blank
Reserved	20	
Owner Name and Address	10	
Blank	29	
	80	Bytes

Foil #16 - #16.1

II K Additional O/S Terms & Explanations

Many additional commonly-used O/S terms with understandable definitions or explanations are contained in a chapter of the IBM manual: IBM System/360 Operating System Operator's Guide: C28-6540.

III A General Organization and Function of O/S

The Operating System is composed of control programs and processing programs. The control programs consist of the following functions:

Data Management

Job Management

Task Management

The processing programs include:

Language Translators

Service Programs

User-Written Problem Programs

Foil #2

III B.1 General Functions and Organization of the DATA MANAGEMENT ROUTINES OF O/S

Function -- Systematically organize, identify, store, catalog and retrieve all types of data and programs. This includes all operations involving the data such as external storage space allocation, channel scheduling and naming, storing, and moving data. These routines are used by both the processing programs and control programs.

Organization -- The DATA MANAGEMENT routines consist of:

(1) Input/Output Supervisor
This supervisor performs all input/output operations while controlling input/output and channel request queues. (Processor of I/O interrupts)

(2) Directory Manager
Maintains an index series called the catalog and locates data sets on secondary storage.

Foil #3

III B.2

(3) Direct Access Device Space Manager
This routine (DADSM) allocates secondary storage space on direct access devices.

(4) Open/Close Routine
This routine performs the required initialization and closing out of all input/output operations.

(5) Access Methods
This group of routines collects the necessary parameters required by the INPUT/OUTPUT SUPERVISOR routine to perform preparatory functions for input/output operations.

Foil #3

III C.1 General Functions & Organization of the JOB MANAGEMENT ROUTINES OF O/S

Function -- Enables a continuous series of jobs to be processed by the system. Primary functions are to read and interpret the O/S JOB CONTROL statements, to initiate the requested job, to terminate completing jobs, and to inform the operator of situations during processing by issuing message on the console.

Organization -- The JOB MANAGEMENT routines consist of:

(1) Master Scheduler

Analyzes commands from the console and handles the printing of messages from the system (from both the control program and problem programs) to the operator.

(2) Reader/Interpreter

Reads the input stream, interprets the control statements, and puts data in the input stream on a direct access device.

Foil #4

III C.2

(3) Queue Manager

Maintains the job queue (that is, lists of job steps in the order in which they are to be performed).

(4) Initiator/Terminator

Controls the allocation of I/O devices and external storage. It initiates the execution of a job and subsequently terminates it.

(5) Output Writer

Writes the output from both system programs and user problem programs.

Foil #4

III D.1 General Functions & Organization of the TASK MANAGEMENT ROUTINES of O/S

Function -- Controls the entire operating system itself by handling the S/360 hardware interrupt function by processing the interrupt, determining its type and branching to appropriate routine for further control. Performing concurrent jobs is primarily a function of the TASK MANAGER. The TASK MANAGEMENT ROUTINES collectively are referred to as the O/S SUPERVISOR.

Organization -- The TASK MANAGEMENT routines consist of:

(1) Interrupt Handlers

These routines analyze the interrupts as they occur.

(2) Contents Supervisor

Maintains a directory of the contents of main storage.

(3) Main Storage Supervisor

Allocates and re-allocates all main storage space.

Foil #5

III D.2

(4) Task Supervisor

Maintains control information for tasks and tasks relationship.

(5) Auxiliary Services

Takes checkpoints when specified, and performs restarts.

Foil #5

III E Processing Programs of O/S

(1) Language Translators

Assembler

Cobol

Fortran

PL/1

Report Program Generator

(2) Service Programs

Linkage Editor

Sort/Merge

Utilities

(3) User-Written Problem Programs

Foil #6

III F Summary of O/S 360

(Summarize the point covered in the preceeding pages of Section III in conjunction with Foil #7.)

Foil #7

IV A O/S Data Set Components on Direct Access

A. On disk, the O/S pack is known as SYS RES (System Residence) and a listing of the VTOC (Volume Table of Contents) will show the following data sets which comprise the operating system:

SYS1.NUCLEUS SYS1.LOGREC SYS1.SVCLIB SYSCTLG	Must be on SYS RES's IPL volume if SYS RES is on more than 1 pack.
SYS1.LINKIB SYS1.SYSJOBQUE SYS1.PROCLIB	
SYS1.MACLIB	May be on SYS RES's IPL volume or on other volume if SYS RES occupies more than 1 pack.
SYS1.MODLIB SYS1.GENLIB	

Foil #18

IV B The O/S System Library & Catalog Facility

The complete operating system is stored, as an organized library, on as many auxiliary storage volumes as required. The library has a built-in catalog that may be used to identify and locate any of the parts of the control program or processing programs. (Additionally, the library catalog enables the programmer to store his own private data as part of the library and later retrieve it by a unique alphanumeric name. The library consists of collections of data called data sets; using this data set name (supplied by the program), the system can search the catalog to determine the location of the data set in the library, and subsequently retrieve it for the program.)

Considering the function of O/S itself, the system is such that NUCLEUS, LOGREC, SVCLIB, and the SYSCTLG itself are always on the IPL volume. When the control program requires the use of its data sets (SYS1.---), the SYSTEM CATALOG is searched to find that data set. The data set is then directly available whether it be on the IPL volume or another on-line volume.

Foil #19

IV C.1 SYS1.NUCLEUS

Used only at IPL time.

Its contents are loaded at IPL.

Core resident after IPL.

Sys 1. Nucleus -- the library containing the nucleus of O/S.

The partitioned data set that contains the resident portion of the control program (i.e., the nucleus)

Foil #20-21

IV C.2 SYS1.LOGREC

Used for System's customer engineer to log data regarding I/O errors.

The data set on disk containing the recorded statistical data on machine errors.

Foil #20-21

IV C.3 SYS1.SVCLIB

Contains non-core-resident routines for:

I/O Error Recovery,

Access methods routines,

SVC routines (e.g., OPEN, CLOSE)

This is the supervisor call library.

The PDS that contains the non-resident SVC routines, non-resident error-handling routines, and the access method routines.

Foil #22-23

IV C.4 SYSCTLG

The system catalog.

Contains pointers for which data set is on which volume -- specific volume numbers are used. Primary use is by job scheduler. This is an optional part of O/S, since volume serial numbers may appear in DD statements rather than the library.

Foil #22-23

IV C.5 SYS1.LINKLIB

This is the location on disk of most O/S program modules that are non-core-resident and system related. The job scheduler resides in LINKLIB.

Master Scheduler resides in LINKLIB

(Many references to LINKLIB w/18K scheduler;
only 1 or 2 references to LINKLIB w/44K scheduler.)

The PDS that contains the IBM-supplied processing programs and part of the non-resident portion of the control program. It may also contain user-written programs which may be brought into core as load modules, and then executed.

Foil #24-25

IV C.6 SYS1.SYSJOBQE

This is a temporary-storage type (or work area) data set used by the JOB SCHEDULER to store information that must be passed from the scheduler to other parts of the control program. O/S elements which refer to SYSJOBQE are:

OPEN, Utility programs, job scheduler, etc.
SYSJOBQE is made up of tables, lists, messages, which is used heavily during scheduler operation -- including storing of what I/O device is allocated for which data set -- and as well at OPEN time.

SYSJOBQE accessed at OPEN time.

SYSJOBQE - Sequential Scheduler Work Queue Area

SYSJOBQE - is a sequential data set

FOIL #24-25

IV C.7 SYS1.PROCLIB

This normally contains cataloged Job Control Language procedures. System reference to the PROCLIB is made during reader-interpreter operation.

PROCLIB is a PDS

Foil #26-27

IV C.8 SYS1.MACLIB

MACLIB is a partitioned data set which contains the definitions (expansions of) the O/S Assembler Language macro-instructions provided by IBM. From this data set, used at assembly time, the Assembler obtains the macro definitions and assembler language statements produced as output from that translator.

Foil #28-29-30

IV C.9 SYS1.MODLIB

This is the source for program modules that are copies or link-edited during systems generation, or updated during program maintenance. Normally kept off-line.

SYS1.GENLIB

These are the macro-instruction definitions unique to the system generation process. Normally kept off-line.

Foil #28-29-30

IV D What Components of O/S Reside Where on SYSRES?

Data Management:

I/O Supervisor	NUCLEUS
Directory Mgr	LINKLIB
DA Space Mgr	LINKLIB
Open/Close Rtn	SVCLIB
Access Methods	SVCLIB or NUCLEUS

Job Management:

Master Scheduler	NUCLEUS/LINKLIB
Reader/Interpreter	LINKLIB
Queue Mgr	NUCLEUS/LINKLIB
Initiator/Terminator	LINKLIB
Output Writer (Sysout)	LINKLIB

Task Management:

Interrupt Handlers	NUCLEUS
Contents Supervisor	NUCLEUS
Main Stor Supervisor	NUCLEUS
Task Supervisor	NUCLEUS
Auxiliary Services	NUCLEUS

Foil #32

V A General Concept of the Core-Resident O/S

Loading of the S/360 Operating System from its SYSTEM RESIDENCE VOLUME (SYSRES) is done by the operator by initiating the standard S/360 load procedure. Several steps follow the pressing of the load key including the loading and initializing of the core-resident portions of the control program. This procedure will be discussed at length in a later section of the course.

Key parts of the control program are loaded into main storage and remain there indefinitely to insure continuous coordinated operation of the system. Other parts of the control program are brought into main storage from auxiliary storage (direct access device(s)) as they are required to perform specific functions.

Foil #33

V B What Does Core Look Like When O/S Is Core-Resident?

Different portions of the control program operate from different areas of main storage. Main storage is divided into two areas: the fixed area, and the problem program (or dynamic) area.

The fixed area of main storage is the lower portion of main storage; its size is determined by the configuration of the control program (based on options selected from the generation of the system -- SYSGEN). The fixed area contains those core-resident control program routines that perform a system function during the execution of a processing program. This fixed area also contains two blocks of main storage for use by transient control program routines; one for transient SVC routines, the other for non-resident I/O error handling routines.

The dynamic area is the upper portion of main storage. It is occupied by a processing program (user's problem program), or control program routines that either prepare job steps for execution (i.e., job management routines) and/or handle data for a processing program (i.e., the access methods).

Foil #34

The O/S Supervisor, the control center of O/S, is composed collectively of those routines discussed under TASK MANAGEMENT: the interrupt handlers, the main storage contents supervisor, the main storage allocation supervisor, and the task supervisor. In its performance of a numerous variety of services for other parts of the control program and all problem programs, the Supervisor functions in the 'fixed area' of main storage, is core resident after System IPL is completed, and is never overlaid by any other O/S routines. A general discussion of the Supervisor is found in Section V E.

The combined efforts of the Scheduler and Supervisor might be viewed in the following manner: the job is read in and begun primarily through the function of the Scheduler; execution of a program is handled primarily by the Supervisor; the completion of a program at End-of-Job Time is handled, once again, by the Scheduler.

Foil #35

 O/S Scheduler/Supervisor Considerations

In viewing the O/S Control Program from a functional point, two distinct sections can be considered: the O/S Scheduler and the O/S Supervisor.

There are basically three different types of schedulers which are available under the general term of O/S: the "Sequential Scheduling System", commonly referred to as Primary Control Program or PCP, are the simplest configurations and functions on the principle of reading jobs (or job steps) sequentially and executing them one at a time; "Sequential Scheduling Systems with Multiprogramming", referred to as OPTION 2 of O/S or Multiprogramming with a Fixed Number of Tasks (MFT), are those which read jobs in sequentially, but can use the 'dynamic area' of main storage for execution of from one to four jobs concurrently; "Priority Scheduling Systems", the most complex scheduling system, is referred to as Multiprogramming with a Variable Number of Tasks (MVT) and permits many jobs to be read in, placed on a queue, and then executed on a priority basis (a maximum of 15 jobs may be executed concurrently). Regardless of the type of Scheduling System used (Selection of which one at SYSGEN) each will consist basically of a MASTER SCHEDULER and a JOB SCHEDULER (discussed in Section V D), function almost entirely in the 'dynamic area' of main storage, and concern itself primarily with automatic job to job transition.

Foil #36

The O/S Scheduler

The Scheduler portion of O/S 360 consists of a MASTER SCHEDULER and a JOB SCHEDULER. The MASTER SCHEDULER (MS) accepts commands from the System Console operator, relays system messages to him, initiates and terminates certain system functions at his request, responds to his inquiries regarding job and system status - in general, it acts as the operator's agent within the system. The MS serves, then, as a two-way communications link between the operator and the system, and in that respect, many of the messages to the operator via the 1052 will have a message prefix of IEE, indicating that they are from the MS. The MS will handle the operator-entered commands to the system: DISPLAY, VARY, START RDR, etc. All messages to the operator via the 1052, whether from the O/S control program or from a problem program are handled by the MS. Under PCP O/S, messages from MS to the 1052 are overlapped by processing but there is no multiple-message buffering so, when a message is sent to the 1052, internal CPU processing continues while the line is typed (WTO - Write To Operator - only; not WTOR - Write To Operator with Reply). However, if a second WTO is given before the first message has completed typing, CPU processing is halted until the first message completes, at which time the second message goes to the buffer and CPU processing continues. Under OPTION 2 of O/S (MFT) multiple-message buffers may be SYSGENed in, which will not halt CPU overlap processing until all available buffers are filled with outgoing messages (WTO only, from several

different problem programs or the control program); a WTOR will cause that partition to cease processing until the message has been replied to (also handled by the MS).

So far as operators are concerned, the MS will differ based on the type scheduler is used (PCP/MFT/MVT) and the commands available to each, for example, under PCP O/S there is no requirement for a SHIFT command (which deals with switching from one partition to another) therefore it does not exist under PCP.

The JOB SCHEDULER (JS) is composed of those Job Management Routines known as the Reader/Interpreter and the Initiator/Terminator and has as its primary functions:

Reading job control statements (JCL) from one or more I/O devices assigned as job input sources;

Allocating I/O devices to each job step;

Initiating the execution of the processing program specified for each step;

Processing selected output produced during each job;

Terminating job steps.

During the functioning cycle of the JS (which is between job steps) the operator may receive 1052 messages with a prefix of IEF indicating that JS was the originator of the message.

Foil #36-#36.1-#36.2

The O/S Supervisor is responsible for the following activities:

- Allocating main storage space required by programs during their execution.
- Loading programs into main storage.
- Controlling concurrent execution of programs.
- Scheduling and controlling I/O operations.
- Providing timing services.
- Providing standard procedures that assist in the diagnosis of exceptional conditions.

The supervisor is available in different versions having different characteristics. Some parts of a version of the control program are required, others are optional and may be included or excluded at system generation time (SYSGEN).

To perform its functions, the Supervisor receives control of the central processing unit by way of an interruption. The interrupt may result from a specific request for services from another part of the operating system or from a problem program (SVC); or it may be an automatic type of interrupt, such as the interrupt that occurs at the completion of an I/O operation (device end, channel end). In this case, that part of the supervisor would be called on to check

that the I/O operation completed successfully (that is, proper channel function, proper control unit and I/O device function, no violation of storage protection, etc.)

Foil #35

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 (done by the SCHEDULER), the control program passes CPU control to a processing program to perform the work required. A processing porgram, 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.

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

When the operator issues a command at the console, the system is interrupted, the supervisor analyses the interrupt, and the command is passed to the MASTER SCHEDULER.

Foil #35

VI A COMMUNICATION MEDIA THE S/360 CONSOLE OPERATOR AND THE OPERATOR SYSTEM

The O/S Control program will communicate with the console operator via:

(1) the 1052 console typewriter for certain system messages and for problem program messages.

(2) the 1403 in cases of messages from some utility programs; at job control statement printing time; at end of job to give disposition of data sets; in the heading of storage dumps; during the processing of some jobs.

(3) a code in the Current Program Status Word (PSW) under conditions of Initial Program Loading error, Supervisor program error, or System environment recording error.

(4) the console alarm bell in certain instances

The S/360 console operator communicates with the O/S control program via:

(1) commands and replies entered on the 1052 console typewriter

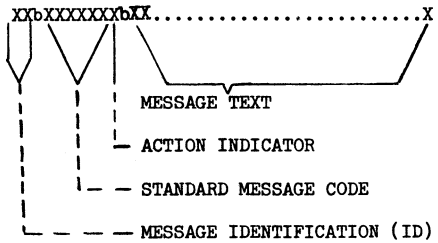
(2) commands entered thru the system input job stream reader device

FOIL #37

VI B.1 1052 MESSAGES FROM O/S TO THE S/360 OPERATOR

O/S Message format:

(exact formats of messages and their meanings are found in IBM System/360 Operating System: Messages, Completion Codes, and Storage Dumps - C28-6631)



FOIL #38

VI B.2 MESSAGE IDENTIFICATION (ID)

A two character numeric field which will appear prior to any message from O/S which requires a reply by the operator. Use this ID number in replying to the message; the reply format is "REPLY XX;YYYY'(EOB)", where XX is the ID number sent from O/S in the initial message to the operator; YYYY is the text of the reply. The control program is going to use the ID number from the reply to determine which message the operator is replying to. (NOTE: It should be pointed out to the operators that this ID can by any number from 00 thru 99, even though during initial use of Primary Control Program O/S it may seem that the ID, and thus the REPLY, is always 00. Impress upon the operators that they must check the ID of the initial message to which they are replying very specifically.)

When the message does not require a reply, the message ID will not preceed the message.

If a job should come to an end before messages requiring replies have been replied to by the operator, a message is written listing all the message ID's which should be ignored by the operator. This message is: 'CANCEL FOLLOWING REPLIES - 00.'

FOIL #39

VI B.3 STANDARD MESSAGE CODE

This is a unique alphameric code, at least six characters long, which is assigned to each different message written by the control program. The code identifies the routine of the control program, or that O/S program that is the source of the message. Refer to foil (FOIL 40) and IBM System/360 Operating System Operator's Guide - C28-6540, for a listing.

FOIL #40 - #41

VI B.4 Action Indicator

The action indicator is a one position alphabetic memonic code that represents the general type of action required. This character prints immediately after the 6 position standard message code. There are five general categories of action wich are defined by these action indicators:

"A" means an "await action condition" which requires the operator to perform a specific action before the task continues.

"D" means an "await decision condition" which requires the operator to choose between alternative courses.

"I" means an "information only condition" which requires no immediate action by the operator.

"W" means an "cannot proceed condition" which stop all processing until a specific action is taken.

"S" means a "cannot proceed condition" which was caused by a hardware malfunction which cannot be corrected by retry. Reception of this type message should be followed by the operator loading the SEREP program.

FOIL #42

VI B. 5 Message Test

This section of the message is separated from the standard message code/action indicator field by a blank. This is a brief statement directed to the operator to supply information, request a specific action, or request information.

FOIL #42

VI C 1052 MESSAGES FROM PROBLEM PROGRAMS

Messages from problem programs are passed to the 1052 via the Master Scheduler. The format of messages to the S/360 Console operator from problem programs via the 1052 is completely free-form, the messages, however, will not exceed one line of print on the 1052. Messages from the problem programs which require the operator's reply (e.g., WTOR-WRITE TO OPERATOR WITH REPLY macro instruction in S/360 assembler language) will be preceeded by the same type message identification (ID) as is used by O/S system messages. This ID will be separated from the text of the problem program message by one blank. As was the case with O/S system messages requiring a reply, this ID must be used by the operator in replying to the problem program message. The format of this type message is, for example: "04 ENABLE 2701, REPLY-UP-WHEN READY." The operator will, when ready, reply with: "REPLY 04, 'UP'(EOB)"

Messages from the problem programs which do not require a reply (e.g., WTO-WRITE TO OPERATOR macro instruction in S/360 assembler language) will not be preceeded by the message identification (ID).

FOIL #43

VI D 1403 Messages from the O/S Control Program

For the most part, messages printed on the 1403 are intended for and directed to the system programmer. There are instances however where messages which can aid the operator are printed on the 1403. Several instances would be: messages which reflect errors in JCL Statements which resulted in the rejection of that particular job for execution by O/S; the printed codes associated with an abnormal program dump presented in an O/S ABEND/ABDUMP; or the printed messages verifying the final data set disposition at job termination time. The extent to which the console operator conerns himself with these types of 1403 messages is dependent primarily on specific organizational policy; it is felt, however, that at a minimum, the operator should have a basic understanding a JCL, its effect on data set disposition, and some exposure to the information presented in a storage dump. (In this course, the topic of JCL and data set disposition is covered in a later section).

In addition to the instances mentioned above, the execution of several O/S service and utility programs can cause messages to the operator to be printed solely on the 1403, or in addition to messages on the 1052. This is the case, for example, with the O/S Sort/Merge program (messages beginning with IER). Messages from the Sort/Merge will not appear on the 1052 console unless that option was specifically requested at system generation (SYSGEN) time.

FOIL #44

VI E.1 DISPLAYABLE PROGRAM STATUS WORD (PSW) CODES TO THE S/360 OPERATOR FROM O/S

In certain error situations, codes are set in the instruction address bytes of the current PSW. This situation can be very serious and is immediately recognizable to the operator by the cessation of all processing and the existence of the system in the WAIT STATE (WAIT light on).

Three general situations can cause this type of circumstance: an error during Initial Program Loading (IPL) or during execution of the Nucleus Initialization Program (NIP); an error detected by the supervisor from which recovery cannot be made; or errors associated with the system trying to recover from a previous error.

(NOTE: These codes and their meaning should now be covered. Refer to IBM manual: IBM System/360 Operating System Messages, Completion Codes, and Storage Dumps-C28-6631 under the section titled: Operator Messages, Wait State PSW Codes.)

In displaying the current PSW according to the procedures for that S/360 model, the operator will find this code in the last three bytes of the PSW. (Review, if necessary, the format of the PSW and/or the method of displaying the PSW on those S/360 models installed.)

FOIL #45

VI E.2

In those circumstances where the error was associated with System Environment Recorder, most operator actions call for the loading and execution of the System Environment Recording Edit and Print (SEREP). (If the operators are not familiar with SEREP, its use should be discussed at this time.)

FOIL #45

VI F USE OF THE CONSOLE ALARM BELL

Under several conditions the console alarm bell on the system will ring, indicating an error condition to the operator. In the case of several supervisor error conditions, the supervisor will attempt to ring the alarm bell before posting the code in the PSW.

Also, in the case an error on the 1052 in printing as message, the alarm bell will ring and reprint of the message will take place.

NO FOIL USED.

VI G.1 OPERATOR COMMANDS TO O/S VIA THE 1052

-GENERAL INTRODUCTION-

Ten commands are available to the S/360 console operator to direct the system in its performance. The formats, functions, parameters, and options of the below listed commands are covered for the Primary Control Program:

CANCEL	SET
DISPLAY	START
MOUNT	STOP
REPLY	UNLOAD
REQ	VARY

(Refer to IBM manual: IBM System/360 Operating System Operator's Guide-C28-6540, for a discussion of each of the commands; foils are included in this course for use in discussion of each command. Space has been allowed each of the following "command" pages for specific instructor notes to be presented over and above those covered in C28-6540).

FOIL #46

VI G.2 OPERATOR COMMAND: CANCEL

FOIL #47

VI G.3 OPERATOR COMMAND: DISPLAY

FOIL #48

VI G.4 OPERATOR COMMAND: MOUNT

FOIL #49

VI G.5 OPERATOR COMMAND: REPLY

FOIL #50

VI G.6 OPERATOR COMMAND: REQ

FOIL #51

VI G.7 OPERATOR COMMAND: SET

FOIL #52

VI G.8 OPERATOR COMMAND: START

FOIL #53

VI G.9 OPERATOR COMMAND: STOP

FOIL #54

VI G.10 OPERATOR COMMAND: UNLOAD

FOIL #55

VI G.11 OPERATOR COMMAND: VARY

FOIL #56

VI H.1 Console Operator Command to O/S Via The System Input Reader

The command statement is used by the operator to enter commands to O/S through the input stream (INPUT READER DEVICE), rather than via the 1052 console typewriter. The general format is:

// in positions (columns) 1 and 2.

b one or more blanks beginning in position 3.

Command

b one or more blands following the command.

Parameters

Valid commands (in the Command field) are:

DISPLAY

MOUNT

SET

START

STOP

UNLOAD

VARY

(NOTE: Cancel, Reply and REQ cannot be entered via the system input reader device.)

FOIL #57

VI H.2

General rules regarding the use of commands in this manner are:

(a) a command statement cannot be continued

(b) the command statement must occur immediately prior to an execute (EXEC) statement, a job (JOB) statement, or a null (//) statement

(c) the command statement must not be interspersed with the input data

(d) the command statement may not take effect during the current job step. (This was also true of certain commands entered via the 1052.)

FOIL #57

VII A OPERATING THE O/S PRIMARY CONTROL PROGRAM (PCP)

-GENERAL CONCEPTS-

The Primary Control Porgram of the Operating System functions in the general manner described in earlier sections of the course (refer to Section V). It functions with a sequential scheduler, allowing one job (or one step of a job) at a time to operate. Normal job to job transition is effected. The PCP Supervisor resides in low main core ('fixed core') and services the problem program resident in upper core ('dynamic core').

FOIL #34

VII B.1 Operating The O/S Primary Control Program (PCP)

-Detailed Operations-

Initial Program Loading (IPL) the Operating System is always the function of the S/360 console operator.

To load the resident portion of the control program, the operator specifies the device address of the system residence initial-program-loading volume in the LOAD UNIT ROTARY DIALS on the S/360 Console. The operator then presses the LOAD KEY on the S/360 Console. A system reset takes place, and then causes an initial-program-load record to be read into core from the specified direct-access device. This is a hardware function of the S/360; this first IPL record (24 bytes) is always found on track zero, cylinder zero of the addressed IPL volume; it is always read into main storage location 0-23. The first eight bytes of this record now is made the current program status word (PSW); the system now begins functioning under program control. The eight bytes at location 8-15 are used as a Channel Command Word (CCW) for the next I/O function. The eight bytes at location 16-23 are also a CCW. The execution of these CCW's cause a second IPL record to be read into main storage. The execution of the second IPL record

FOIL #58 - #59

causes the initial-program-load program to be brought into main storage. Control is given to the IPL program which searches the volume label of the IPLed volume to locate the address of the volume's volume table of contents (VTOC). The VTOC is then searched for the partitioned data set known as SYS1.NUCLEUS, which contains the resident portion of the control program. The nucleus is brought into the 'fixed area' of main storage, and the Nucleus Initialization Program (NIP) which is also part of SYS1.NUCLEUS, is brought into the 'dynamic area' of main storage. NIP receives CPU control from the IPL program and initializes or processes the nucleus; amongst other functions, NIP determines the number of core storage bytes available in this S/360 and checks the interval timer. If the timer is disabled, the message: "IEA100A TIMER IS NOT WORKING." is received on the console; if the timer is enabled, the message is not received. (It might be worthwhile to use this message as a tool if IPLing the system is failing and it is desired to know how far in the IPL procedure the system is proceeding before failure.)

FOIL #59 - #60

If an error other than an I/O error occurs during the running of NIP, the wait light turns on and a message will be typed on the 1052 describing the error. If an I/O error takes place during the IPL procedure or during the execution of NIP, the WAIT light is turned on and an error code is available by displaying the current program status word (PSW); no message is received on the 1052. A listing of these PSW codes and appropriate action is found in the IBM manual: IBM System/360 Operating System Messages, Completion Codes, and Storage Dumps: C28-6631 under the Section titled: Operator Messages Wait State PSW Codes. (This group of codes was studied under the O/S Messages section of this course). Note that in most cases re-IPLing should be done and if successive retries fail, the IBM CE should be notified. If this is the case, be sure to inform the CE what the code and explanation are from the O/S manual. After NIP completes its processing, it causes itself to be replaced in the 'dynamic' area of main storage by the master command routine of the master scheduler and the reader/interpreter, which are read in from SYS1.LINKIB.

Control now passes to the master scheduler which issues the message 'IEE007A READY." and wait light will turn on. (From Chapter VI of this course, it can be seen that the Master Scheduler has issued this message and that the action indicator part of the message is "A" signifying that the operator must take some positive action.) Since this is

FOIL #60 - #61 - #62

the first "ready" message received by the operator, this is when the operator must set the date, done by typing on the 1052: "SET DATE=YY.DD(EOB)", where YY is the year and DDD is the day of the year (Julian date) expressed in three numeric digits. If the operator does not set the date at this point, the message: "IEE013I REJECTED. SET DATE REQUIRED." will be received on the 1052, in which case the date should now be set, as indicated above. The operator may also specify in this SET command:

(1) If the system includes the timer option--- "SET DATE=YY.DD, CLOCK=HH.MM.SS (EOB)", where HH is the hour of the day, MM is the minute, and SS is the second.

(2) If the job control statements disk data set SYS1.PROCLIB is on other than the SYSRES volume, that other volume's device address may be specified--"SET DATE=YY.DDD,CLOCK= HH.MM.SS,PROC=PPP(EOB)" where PPP is the S/360 device address, such as 191.

FOIL #63

(3) If an input work queue (SYSJOBQE) is not on SYSRES--"SET DATE=YY.DDD,CLOCK=HH.MM. SS,PROC=PPP, Q=QQQ(EOB)", where QQQ is the S/360 device address of the unit which has the volume containing the queue.

(4) If the SYSJOBQE is to be formatted--"SET DATE=YY.DDD,CLOCK=HH.MM.SS,PROC=PPP,Q=(QQQ,F) (EOB)", where F is the letter F (upper case). (NOTE: the queue is normally formatted only the first time the system is IPLed after building the system (SYS GEN).

If at SYSGEN time, your installation selected an automatic option for system input reader (RDR) and the system output writer (WTR), the next two message will now print on the 1052:

START RDR, unitname 1

START WTR, unitname 2

(NOTE: Normally unitmane 1 is ØØC, the system card reader; unitname 2 is normally ØØE, the system printer. These devices, however, could be any appropriate input or output device, respectively. Or, if the input reader is normally from ØØC, but for this job or series of jobs a tape has been prepared as input, the operator may enter the command:

FOIL #64

"START RDR,TTT", where TTT is the address of the tape drive now having the input job stream. This command will override the previous message designating the input reader. The same procedure holds true from the output writer.

Following the printing of the START RDR and START WTR commands, the operator may now set some devices to an offline capcity if he so desires. This is done, for example, by typing:

"VARY 193, OFFLINE(EOB)"

It is the master scheduler who will handle this. The next command is performed by the operator. At this point he types: "START(EOB)" on the 1052. This causes the master scheduler, to pass control to the reader/interpreter, in 'dynamic' core, which now begins to read job control statements from the device specified in the START RDR command previously issued. The reader routines accept what is being brought in from the input reader device and turns that information over to the interpreter routines which translates the job control languaage statements into meaningful functions to be performed. The interpreter's translations are put into the SYSJOBQE, an O/S data set on disk.

(At this point brief description of JCL is in order)

FOIL #63 - 66

 Job Control Language Statements (JCL)

The following pages discuss JCL in a rather general form; the foils however do treat specifies and in some cases examples are used. It is intended only as a guide; the depth to which the JCL terms should be covered depends entirely on the specific operations department policy on what each level of operator should know.

Job control statements are of primary importance to the programmer; they are the means by which the system is told what job this is, what program is to be executed, and what data sets are to be used in the execution of the program. It is these statements which are read by the reader routines of the job scheduler and translated into functions to be performed by the interpreter routines.

There are six general types of job control statements:

(1) Command Statement
(2) Job Statement
(3) Execute Statement
(4) Data Definition Statement
(5) Delimiter Statement
(6) Null Statement

FOIL #67

(1) The command statement is used to enter the normal O/S commands available to you, the operator, into the system by means of an input device, as opposed to the normal method, via the 1052 console typewriter. The commands, as covered in a previous section of this course, must be preceeded by two slashes "//" followed by at least one blank.

Examples are shown on Foil #72

(2) The job statement is the first statement of a job (only command statements may preceed it). It indicates to O/S the beginning of a new job and the end of the previous job. This statement normally contains information such as the job name, an account number, the programmer's name, and directions as to whether or not to print the JCL statements on the printer. Several fields are always required; many optional fields may exist.

Examples are shown on Foil #68 - #68.1

(3) The execute statement is the first statement of each job step within the job. (A job may be made up of one or many steps; each step is reflected by an execute statement.) This statement usually contains the name of the step, the name of the program to be executed, additional parameters to be passed to the program, etc. Several fields are required and many are optional. (Command statements or a JOBLIB DD statement may precede the execute statement.)

Examples are shown on Foil #69 - #69.1

(4) The data definition (DD) statement describes to the operating system what data sets are to be used in the execution of this job step. Many fields are available and the actual DD statement may be continued on several consecutive (continuation) records (e.g. cards) in such a case. Information normally contained includes the name of the data set to be used, perhaps its device type (e.g. 2311), or a reference to this data set's volume serial number. Fields are also used to identify to O/S that this data set is to be newly created during execution of this job step, or perhaps it already exists and the system is to use the 'old' data set. Disposition of the data sets at end of job time is also explained: 'keep' this data set, disgard ('delete') it, or perhaps catalog it into a library.

Examples are shown on Foil #70 - #71

(5) The delimiter statement (simply a '/*' in columns 1 and 2) is used to separate data (e.g., object decks or input data) in the input stream (intermixed with JCL statements coming into the system through the input reader device) from the JCL statements which follow this data. This statement is required at the end of every data set coming into the system via the input stream (input RDR).

FOIL #72

(6) The null statement (simply '//' in columns 1 and 2) is used to mark the end of the last job in the input stream coming into the system through the input reader device. (On the card reader, this card signals end-of-file.) (Needless to say, this statement should not appear within a job or job steps or it will act as a terminator of information for the current job.)

FOIL #72

(7) Examples of JCL Summarization are shown on Foils #73 - #74.

VII B.9

As we left off in the sequence of events, the JCL statements for the first job and first step of that job have been read into the system, the statements have been translated, and the findings have been put on SYSJOBQE; all this was done by the reader / interpreter who resides now in the 'dynamic' area of core.

The reader / interpreter now has itself replaced in 'dynamic' core by the initiator / terminator routines, which are brought in from SYS1.LINKLIB. Control passes to the initiator which locates any input data sets required for execution by using the information translated from the DD cards (or, if the data set required is a cataloged data set, by a catalog search. If this is the case, the search is performed by a catalog management routine that is entered from the initiator.) Since a job step cannot be initiated unless there are enough I/O devices to fill its needs, the initiator determines whether the required devices are available, and then makes specific assignments. If necessary at this point, messages to the operator direct the removal and / or mounting of volumes. Examples of the types of messages which might be received at this time are:

IEF 233A M 180, SCRTCH, TAPRINT
IEF 504A MOUNT THE FOLLOWING VOLUME SERIALS FOR AVR.
RES 002

FOIL # 75 - #76

VII B.10

After I / O devices are assigned, the initiator allocates direct access storage space on volumes for output data sets. To perform this function the initiator calls on the direct access device space manager routines. DADSM, upon finding space available on a specific volume for a specific data set, will make an entry in that volume's table of contents (VTOC) (from an earlier section we know that these entries in the VTOC are DSCB's or data set control blocks and they tell the name of the data set and where it is on the volume). If the volume where the space is required (specified to DADSM by the initiator), does not have enough room for this data set's space allocation, DADSM will return control to the initiator which will issue a message to the operator to mount a new volume or suspend initiating this job step. An example of this message is:

IEF 244I UNABLE TO ALLOCATE FROM AVAILABLE DEVICES.

FOIL #75 - #76

VII B.11

The initiator again is given control and perform its final function - the acquisition of non-direct-access space. The initiator now turns control of the CPU over to the nucleus' routines (e.g. Task Manager) and causes the specified problem program to be fetched into 'dynamic' core, replacing the initiator/terminator routines.

VII B.12

The problem program now begins execution in 'dynamic' core; the problem program issues an OPEN macro for each data set that it will be using, I/O operation are performed by the supervisor when that service is requested by the problem program, interrupts are generated and processed, transient control program routines are brought into 'fixed' core as required from SYS1.SVCLIB, control of the CPU changes hands many times between the problem program executing in 'dynamic' core and the control program executing in 'fixed' core. Finally, the problem program approaches end-of-job and issued a CLOSE macro to each data set it has previously OPENED; the problem program now issues a RETURN macro (for example) which causes control of the CPU again to be returned to the control program. The supervisor now uses the OPEN/CLOSE/EOV routines to CLOSE any data sets left unCLOSED by the problem program.

FOIL #77 - #78

(Had this step gone to an abnormal termination (e.g. because of a program check interrupt) a storage dump would have been provided.)

The supervisor now causes the initiator/terminator routines to be brought into the 'dynamic' area of storage from SYS1.LINKLIB; the reader/interpreter replaces the user's problem program and now performs functions required to terminate individual job steps and/or complete jobs. The terminator gets information from SYSJOBQE and releases the I/O devices, and disposes (gives DISPOSITION for) of data sets used and/or created during this job step. The terminator will give final disposition of data sets according to what the programmer specified regarding each data set in the DD statements of his JCL statements (e.g., KEEP, DELETE). If the JCL statements for the next job step have already been read and interpreted, the initiator again takes over at this point for the execution of this step. If the JCL statements were not read previously, the initiator/terminator is replaced in 'dynamic' core by the reader/interpreter (from SYS1.LINKLIB) which begins once again the read-interpret-initiate-execute-terminate cycle for the next job.

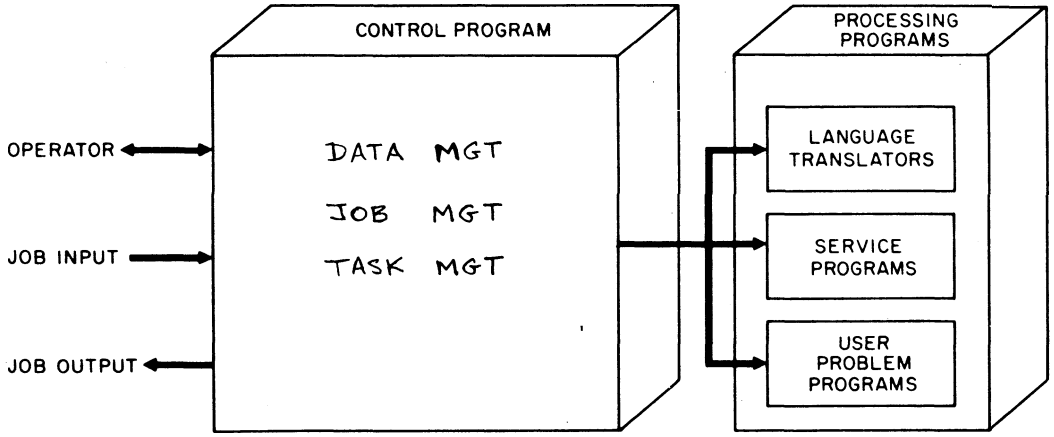
FOIL #79 - #80 - #81

OBJECTIVES OF AN OPERATING SYSTEM

- INCREASE THRUPUT
- DECREASE TURN AROUND TIME
- ASSIST INST MANAGER
- ASSIST PROG OF APPLICAT
- ASSIST SYSTEM OPEREATOR

#1

GENERAL ORGANIZATION & FUNCTION OF THE OPERATING SYSTEM



DATA MANAGEMENT RTNS

FUNCTION- ORG, IDENT, STORE CATALOG, & RETRIEVE DATA.

ORGANIZATION:

1- I/O SUPERVISOR
2-DIRECTORY MGR
3-DASD-MGR
4-OPEN/CLOSE RTN
5-ACCESS METHODS

JOB MANAGEMENT RTNS

FUNCTION: ENABLE CONT SERIES OF JOBS TO BE PROC BY THE SYSTEM.

ORGANIZATION:

1- MASTER SCHEDULER

2- READER/INTERPRETER

3-QUEUE MGR

4-INITIATOR/TERMINATOR

5-OUTPUT WRITER

TASK MANAGEMENT RTNS

FUNCTION: INTERFACE WITH S/360 TO HANDLE INTERRUPTS; CONTROL TASKS CONTENDING FOR EXECUTION

ORGANIZATION:

1- INTERRUPT HANDLERS

2-CONTENTS SUPERVISOR

3-M.-S. SUPERVISOR

4-TASK SUPEPVISOR

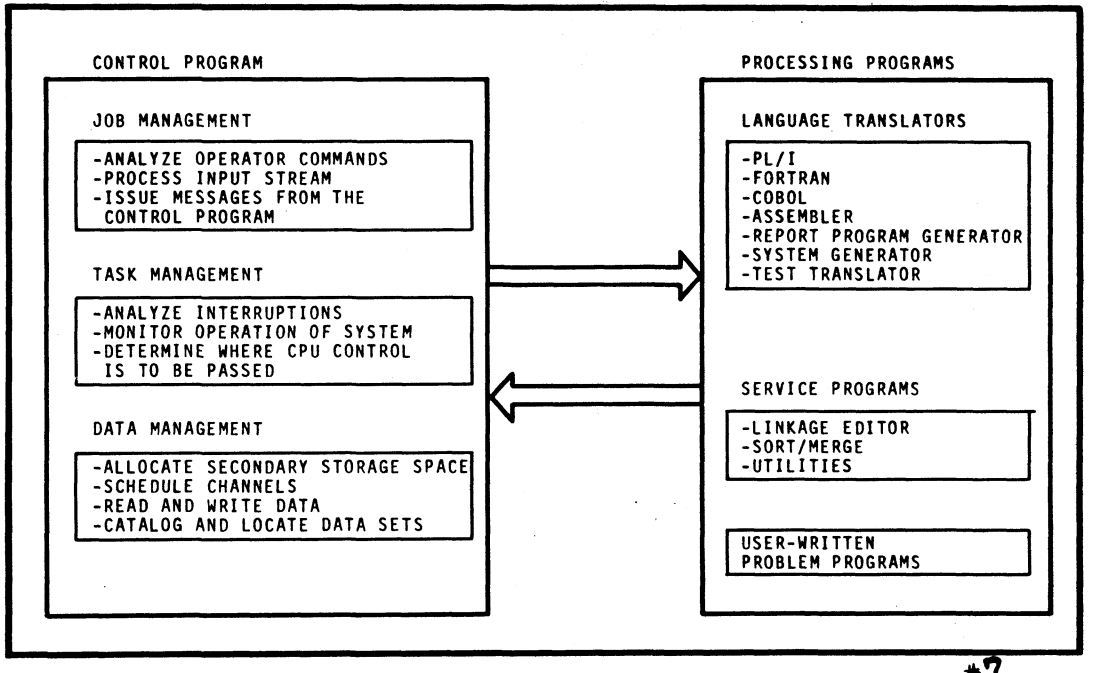
5-AUXILIARY SERVICES

PROSESSING PROGAMS - O/S

- LANGUAGE TRANSLATORS
  - ASSEMBLER
  - COBOL
  - FORTRAN
  - PL/I
  - RPG
- SERVICE PROGRAMS
  - LINKAGE EDITOR
  - SORT/MERGE
  - UTILITIES
- USER-WRITTEN
  - PROBLEM PROGRAMS

*6

OPERATING SYSTEM/360



*7

VOLUME

- *REEL OF TAPE*
  2400 MAGNETIC TAPE
  7340 HYPERTAPE
- *DISK PACK*
  2311, 2314 DISK STORAGE UNIT
- *DATA CELL*
  2321 DATA CELL DRIVE
- *ACCESS MECHANISM*
  2302 DISK STORAGE DEVICE
- *2301 DRUM STORAGE*

#8

DATA SET NAMES

SIMPLE

A
A2
A1234567
INVENTRY

QUALIFIED

E.A.P
INVENTRY.LOC695.PARTNO25
A.B.C.D.E.F.G.H.I.J.K.L.M.N.O.P.Q.R.S.T.U.V
TREE.FRUIT.APPLE

#9

JOB CONTROL LANGUAGE

TELLS THE CONTROL PROGRAM-
- WHAT PROGRAM TO EXECUTE,
- WHAT THE JOB NAME IS,
- WHAT DATA SETS ARE REQUIRED,
- WHERE THOSE DATA SETS ARE, ETC.

JCL STATEMENTS -

- JOB
- EXECUTE
- DATA DEFINITION
- COMMAND
- DELIMITER
- NULL

JOB AND JOB STEPS

JOB - INCLUDES 1 OR MORE STEPS (PROGRAMS) TO BE EXECUTED

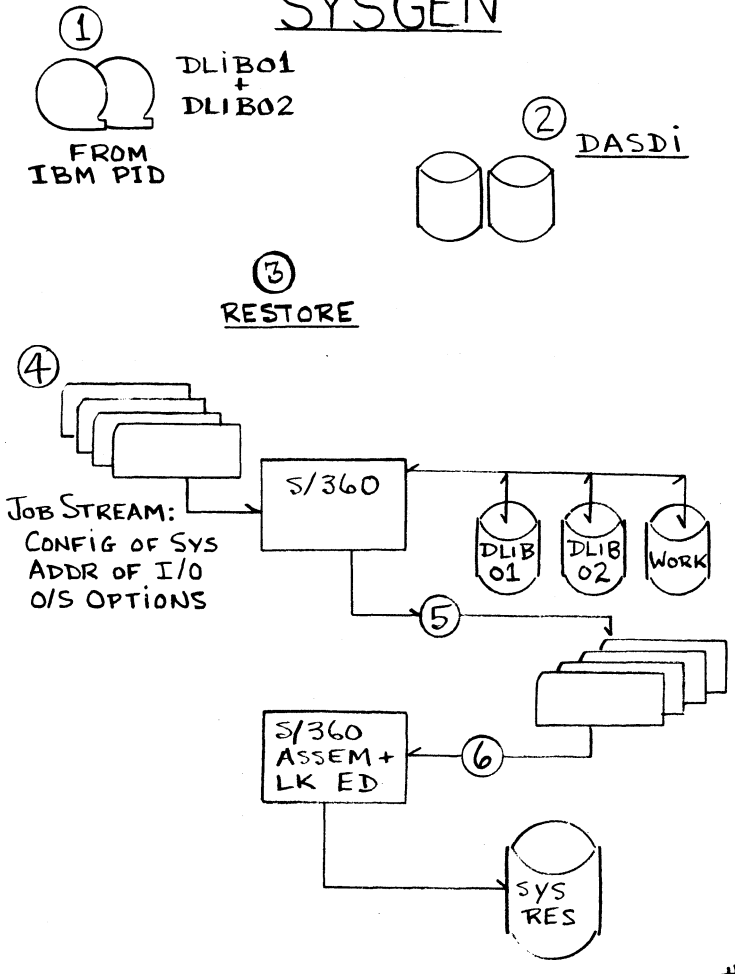
STEP - 'EXECUTE' STATEMENT; A SINGLE PROGRAM TO BE EXECUTED.

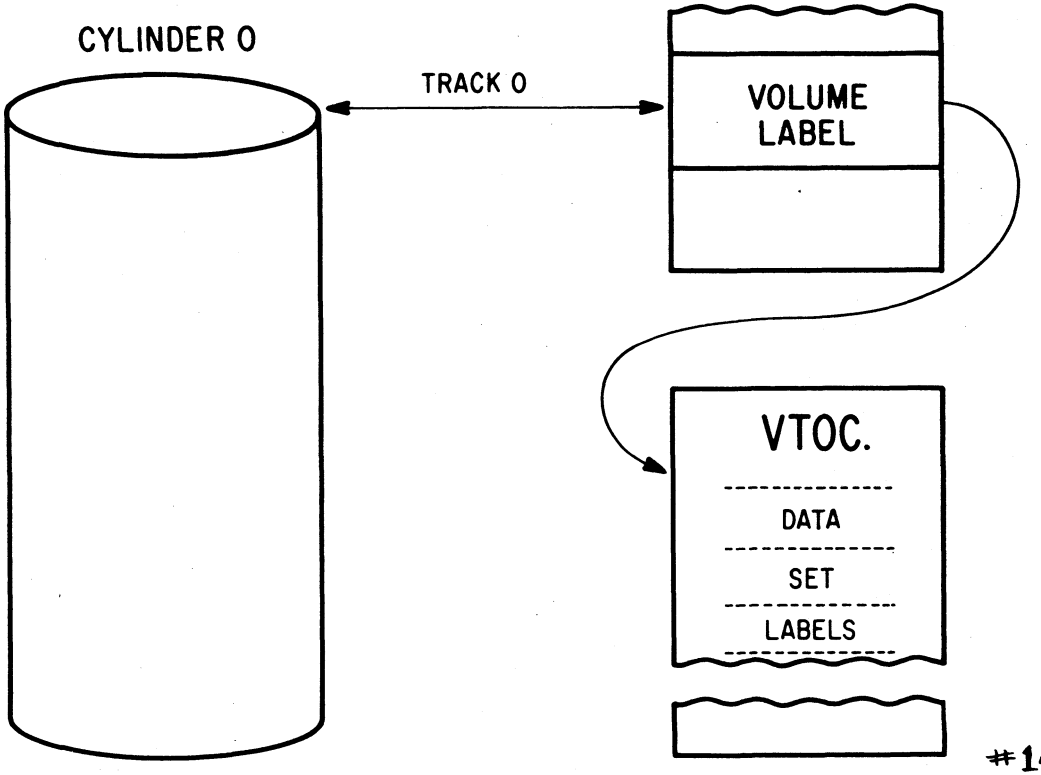
JOB: FORTRAN

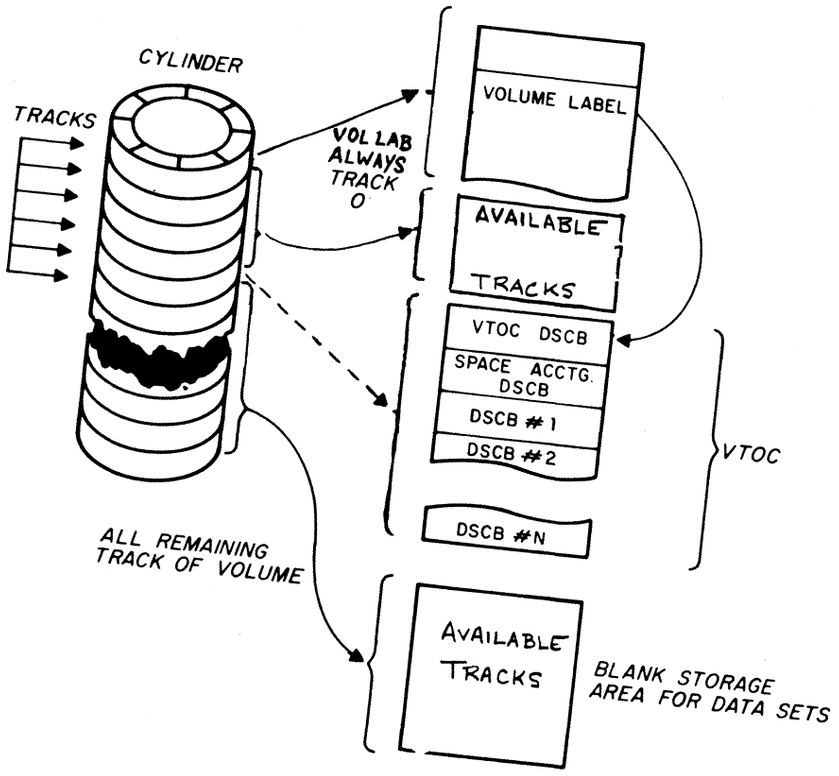
STEP 1: COMPILE FORTRAN SOURCE
STEP 2: LINK EDIT FORTRAN OBJECT
STEP 3: EXECUTE FORTRAN

SYSTEM GENERATION

SYSGEN



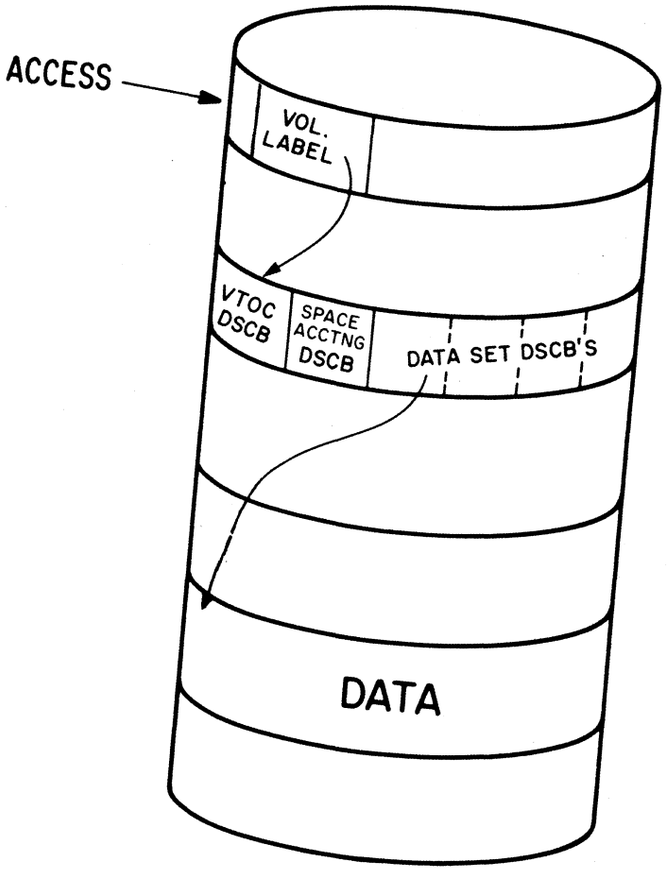




#15

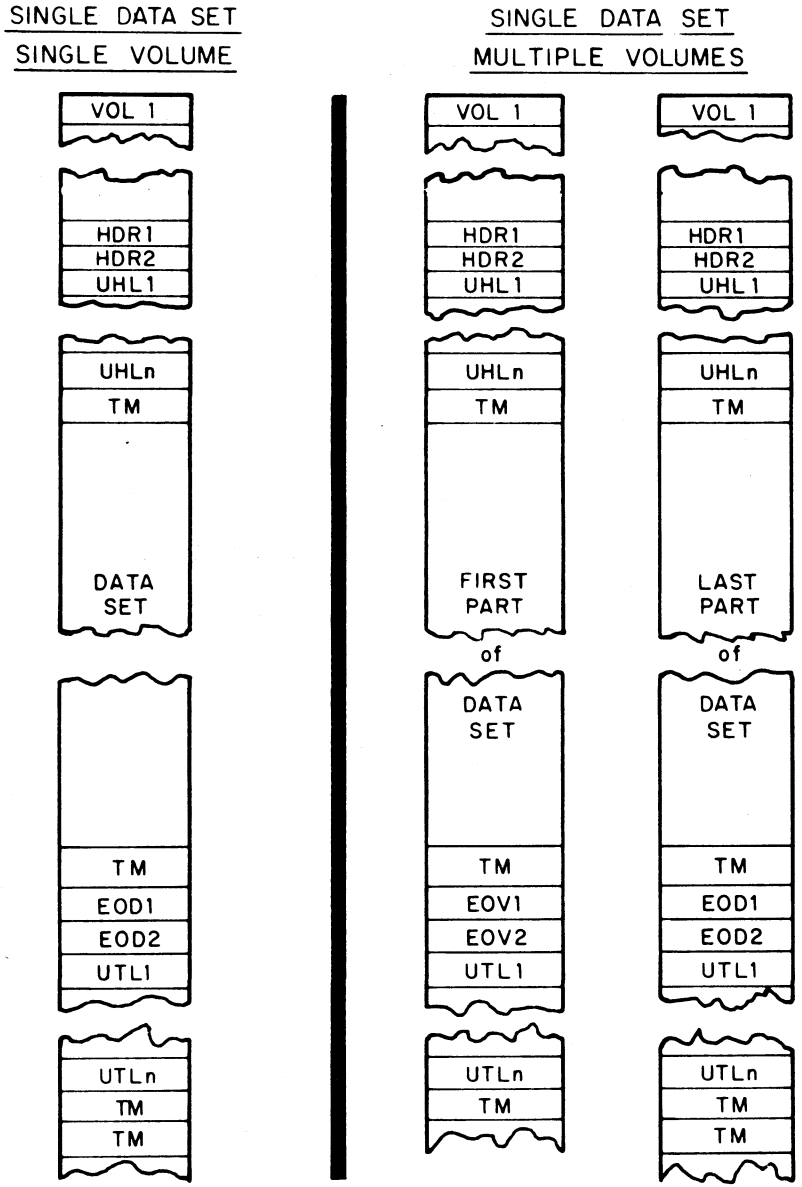
DASD LABELS

ONLY STANDARD LABELS ARE USED



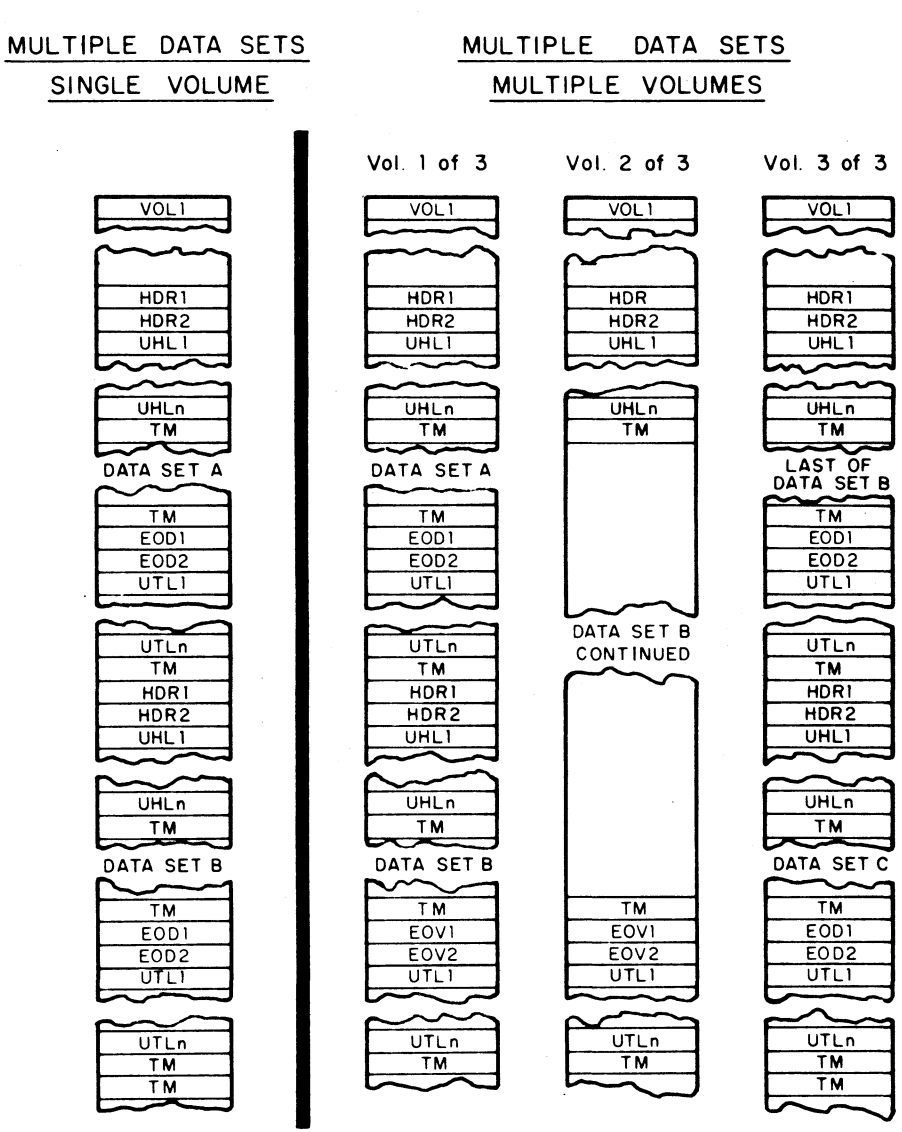
#15.1

STANDARD LABEL FORMATS FOR MAGNETIC TAPE



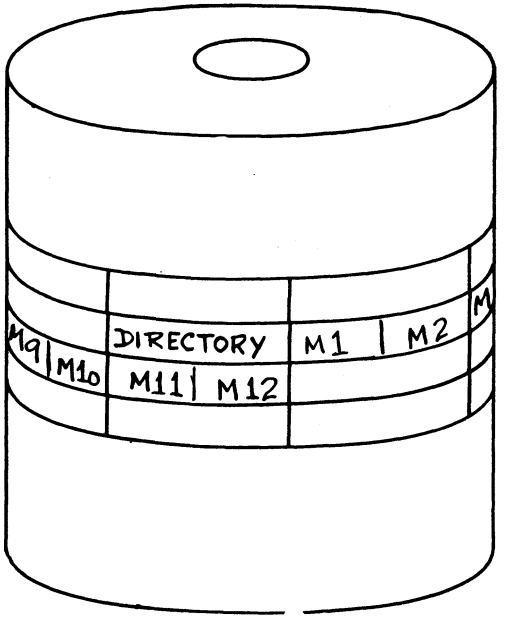
#16

STANDARD LABEL FORMATS FOR MAGNETIC TAPE



#16.1

PDS ORGANIZATION



O/S COMPONENTS ON DASD

SYS1.NUCLEUS
.LOGREC
.SVCLIB
SYSCTLG

→ MUST BE ON IPL VOLUME

SYS1.LINKLIB
.SYSJOBQE
.PROCLIB

SYS1.SYSUT1
.SYSUT2
.SYSUT3

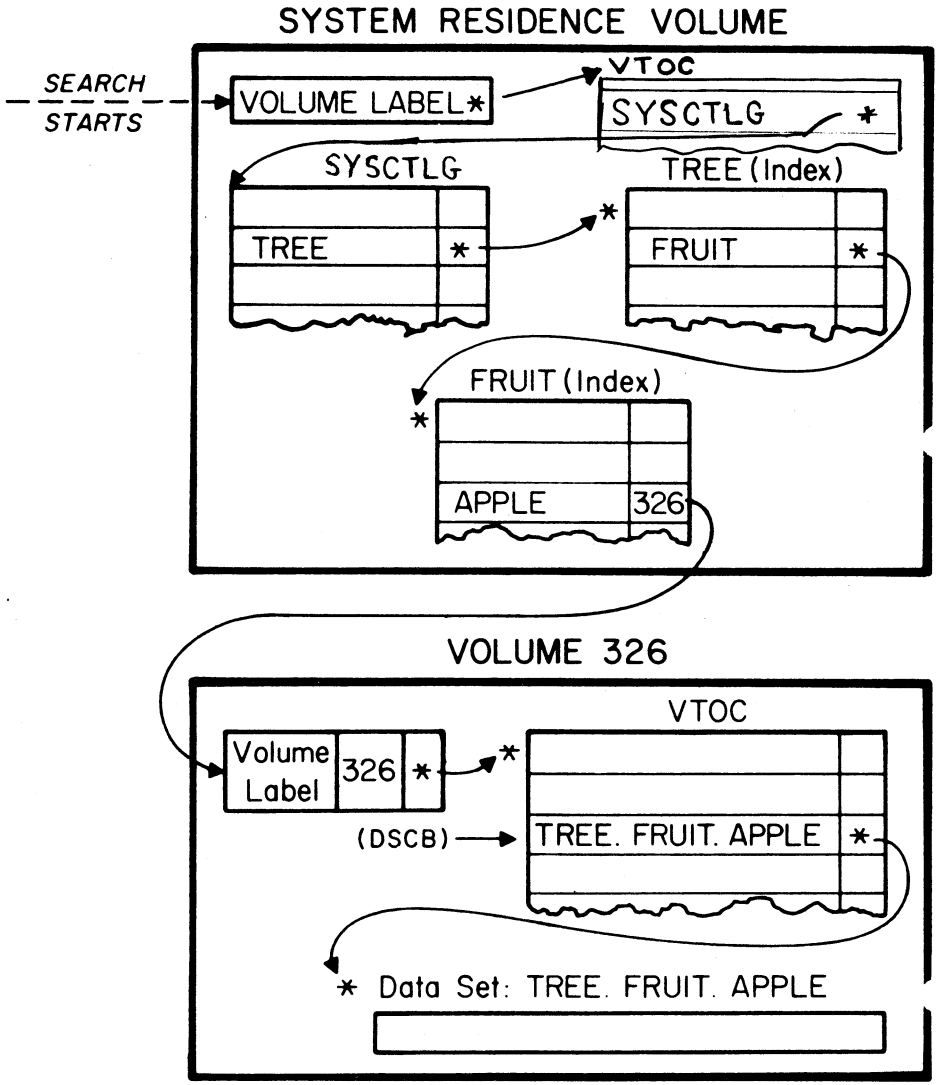
SYS1.MODLIB
.GENLIB

SYS1.MACLIB

→ MAY BE ON OTHER THAN IPL VOLUME

CATALOG SEARCH

FIND: DATA SET TREE.FRUIT.APPLE



#19

SYS1.NUCLEUS

A PARTITIONED DATA SET

CONTAINS CORE-RESIDENT PORTION OF THE CONTROL PROGRAM - THE NUCLEUS

LOADED BY IPL; CORE-RESIDENT AFTER LOADING

SYS1.LOGREC

A SEQUENTIAL SYSTEM DATA SET

USED BY O/S TO LOG SYSTEM/CHANNEL MALFUNCTIONS + FAILURES.

VALUE TO IBM FEs

#20-21

SYS1.SVCLIB

A PARTITIONED DATA SET

CONTAINS NON-CORE-RESIDENT RTNS FOR-
- I/O ERROR RECOVERY
- ACCESS METHODS
- SVC ROUTINES (OPEN, CLOSE, ETC.)

SYSCTLG

THE SYSTEM'S CATALOG

CONTAINS POINTERS TO ALL
SYS1.xxx TYPE DATA SETS

SYS1.LINKLIB

A PARTITIONED DATA SET

CONTENTS INCLUDE:

MASTER SCHEDULER, JOB SCHEDULER,
AND IBM-SUPPLIED PROCESSING PROGS:
ASSEMBLER - LINKAGE EDITOR -
FORTRAN COMPILER, ETC.

MAY ALSO CONTAIN USER'S PROGRAMS

SYS1.SYSJOBQE

SEQUENTIAL SCHEDULER WORK QUEUE AREA

TEMPORARY STORAGE TYPE DATA SET

USED BY JOB SCHEDULER TO STORE
INFORMATION ABOUT A JOB

COMPOSED OF MANY TABLES, LISTS,
AND MESSAGES.

SYS1.PROCLIB

A PARTITIONED DATA SET

CONTAINS JOB CONTROL STATEMENTS (ENTIRE PROCEDURES) WHICH HAVE BEEN CATALOGED.

JCL CAN BE CALLED IN FROM PROCLIB.

SYS1.SYSUT1,2,3

PROCESSING PROGRAMS' WORK AREAS

TEMPORARY - SCRATCH - TYPE DATA SETS

USED, FOR EXAMPLE, BY:

ASSEMBLER (1-2-3)

LINK EDIT (1)

SYS1.MODLIB

USED AT SYSGEN

NORMALLY NOT ON-LINE

SYS1.GENLIB

USED AT SYSGEN

NORMALLY NOT ON-LINE

SYS1.MACLIB

USED BY ASSEMBLER

CONTAINS CODING GENERATED BY ASSEMBLY-LANGUAGE MACROS

JOBLIB

A PARTITIONED DATA SET
USED BY O/S IN SAME MANNER AS LINKLIB
LIBRARY OF USER'S PROBLEM PROGRAMS
WHICH MAY BE LOADED DIRECTLY
FROM DASD AND EXECUTED.

O/S ON SYSRES

DATA MGT-

I/O SUPERVISOR - NUCLEUS
DIRECTORY MGR + DA SPACE MGR - LINKLIB
OPEN/CLOSE RTNS - SVCLIB
ACCESS METHODS - SVCLIB OR NUCLEUS

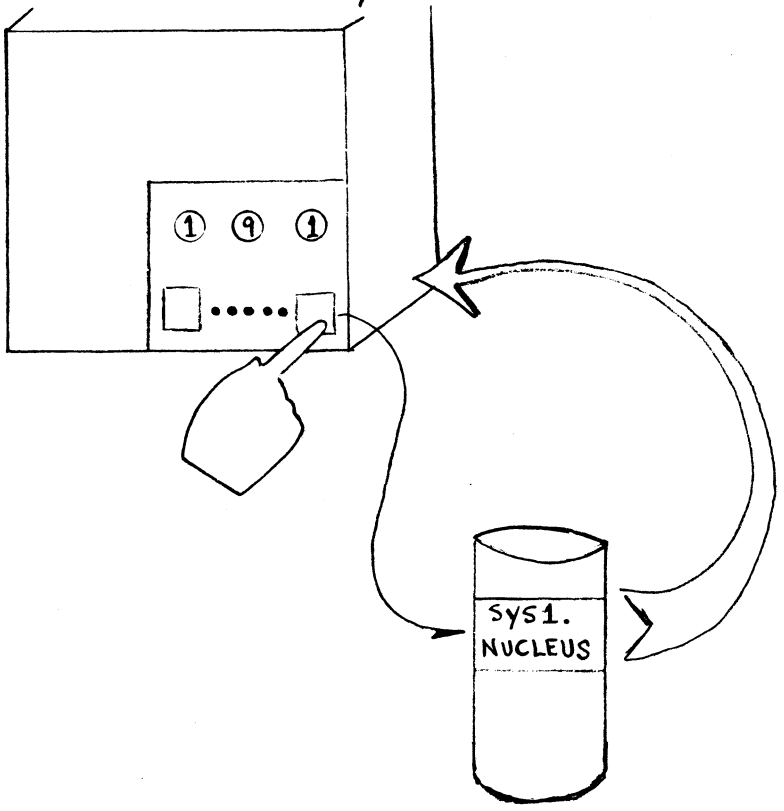
JOB MGT-

MASTER SCHEDULER - NUCLEUS/LINKLIB
READER/INTERPRETER, OUTPUT WRITER,
+ INITIATOR/TERMINATOR - LINKLIB

TASK MGT-

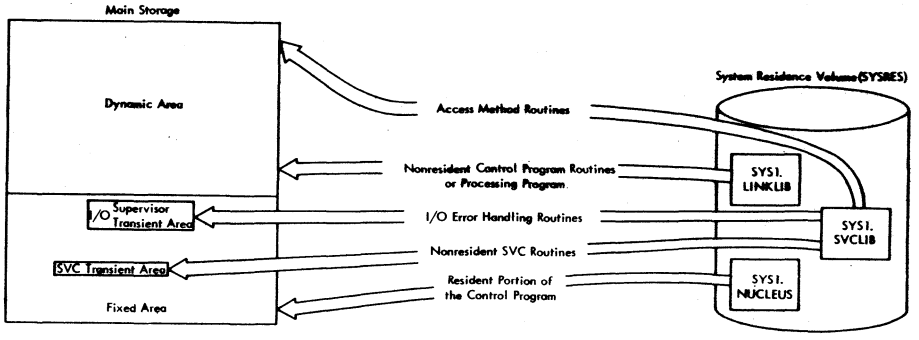
INTERRUPT HANDLERS
CONTENTS SUPERVISOR
MAIN STORAGE SUPERVISOR
TASK SUPERVISOR
AUXILIARY SERVICES
} NUCLEUS

LOADING THE O/S CONTROL PROGRAM



191
SYSRES

CORE-RESIDENT PRIMARY CONTROL PGM



O/S SUPERVISOR:

- CONTROL CENTER OF ENTIRE O/S
- INTERFACES WITH HARDWARE
- RESPONSIBILITIES:

  LOAD PROGRAMS
  ALLOCATE MAIN STORAGE DURING EXECUTION
  CONTROL CONCURRENT PGM EXECUTION
  SCHEDULE + CONTROL I/O OPERATIONS

SCHEDULERS

- SEQUENTIAL SCHEDULER
- SEQUENTIAL SCHEDULER WITH MULTIPROGRAMMING
  (O/S OPTION 2 -
  PARTITION STORAGE -
  MFT: MULTIPROGRAMMING WITH A FIXED NUMBER OF TASKS.)
- PRIORITY SCHEDULER
  (O/S OPTION 4 -
  MVT: MULTIPROGRAMMING WITH A VARIABLE NUMBER OF TASKS.)

SCHEDULER COMPONENTS

MASTER SCHEDULER -

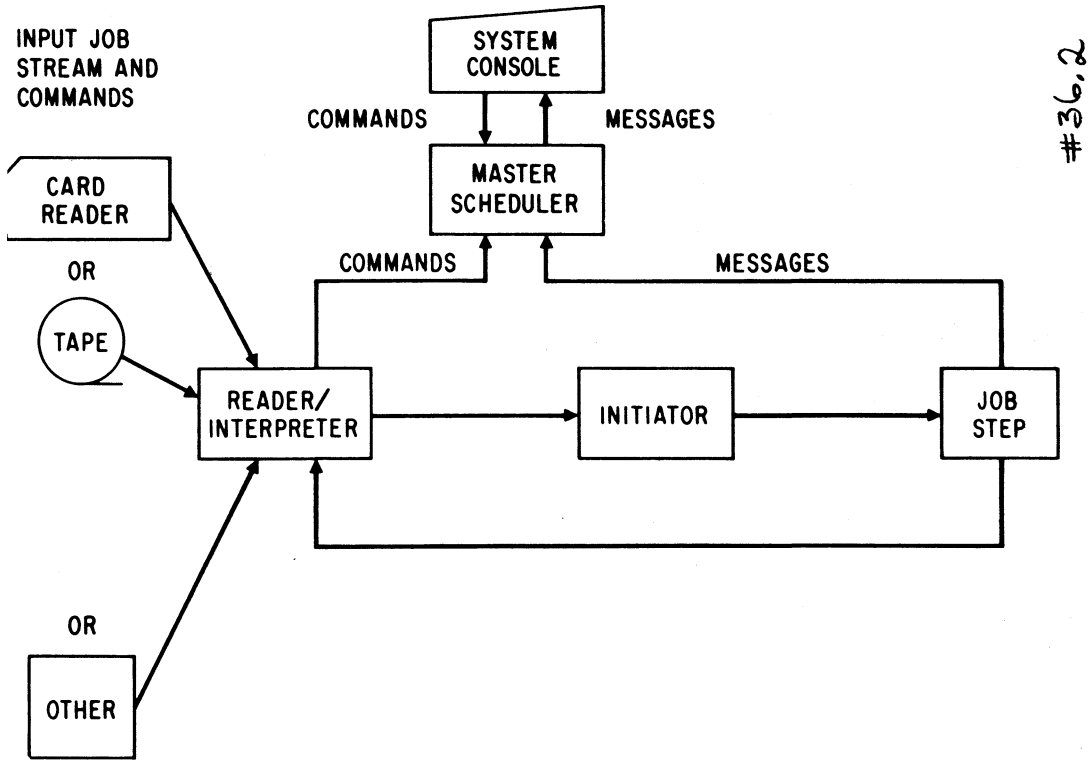
ACCEPTS COMMANDS FROM OPERATOR,
SENDS MESSAGES TO OPERATOR,
CALLS FOR JOB SCHEDULER, WHEN NEEDED.

JOB SCHEDULER -

R/I + I/T = JOB SCHEDULER
READS JCL
ALLOCATES I/O FOR JOBS
INITIATES EXECUTION
HANDLES JOB TERMINATION - DISPOSITION

#36.1

SEQUENTIAL SCHEDULING SYSTEM



#36.2

OPERATOR ⟷ S/360 O/S COMMUNICATIONS

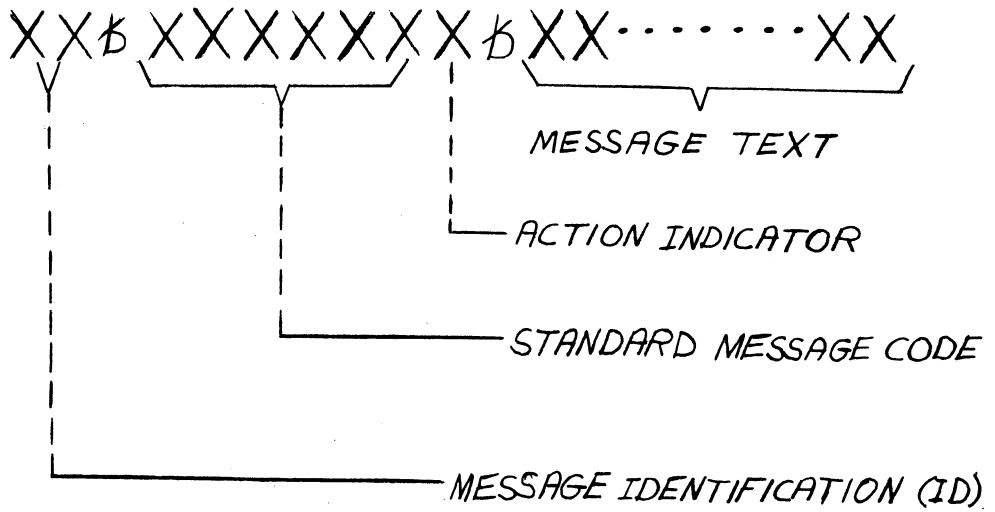
To OPERATOR:

- 1052 CONSOLE MESSAGES
- 1403 PRINTER MESSAGES
- WAIT STATE PSW CODES
- CONSOLE ALARM BELL

FROM OPERATOR:

- 1052 CONSOLE COMMANDS AND REPLIES
- SYSTEM INPUT READER COMMANDS

O/S MESSAGE FORMAT



O/S MESSAGE ID

00 IEF247I REPLY 3 CHAR DEVICE NAME OR CANCEL

ID

REPLY 00,'CANCEL' (EOB)

04 IEC107D E 192,RESADS,OUTPUT1

REPLY 04,'U' (EOB)

O/S MESSAGE SOURCES

IBC	INDEPENDENT UTIL
IEA	SUPERVISOR
IEC	INPUT/OUTPUT
IEE	MASTER SCHED
IEF	JOB. SCHED
IEH	SYS/SUPPORT UTIL
IEM	PL/1
IEP	COBOL
IEQ	COBOL
IFB	I/O ERROR RECORDING
IFBF	SYS ERROR RECORDING
IFC	INITIALIZE LOGREC
IHC	FORTRAN SUBRTNS
IEU	ASSEMBLER

STANDARD O/S MSG CODE

IEA000A INT REQ 00C 00 0200 4000

IEA000I I/O ERR 190 86 0E00 0800
00090007

IEC101A M 193,RES004

IEE007A READY.

IEF101I RDR CLOSED

IEH506I NO ACTION TAKEN

IEV999 ASSEMBLY TERMINATED. PERMANENT I/O ERROR ON SYSUT1.

IFB003I SER RECORDING AREA FULL

IHC001A PAUSE 00001

ACTION INDICATOR

ONE POSITION ALPHA CHARACTER:

A IEA100A TIMER IS NOT WORKING.

D IEC007D E 192,RES003,PREM4

I IEC004I D 193,CX3333

W IEF401W I/O ERROR SCHEDULER WORK AREA

S IFB708S MACHINE ERROR EXECUTE SEREP

PROBLEM PROGRAM MESSAGES

00 ADS- ENABLE 2701. REPLY WITH WORD-UP- WHEN ENABLED.

(REPLY 00,'UP' (EOB))

ADS- DISK SHOWS UNPUNCHED RECORDS. LOAD ADSPUNCH PROG.

EOJ- DEBIT DATA SET CREATED.

1403 O/S MESSAGES

```
//DSPLAY JOB ,MSGLEVEL=1
//DISVTOC EXEC PGM=IEHLIST
//SYSABEND DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SYSRES DD DISP=OLD,VOLUME=SER=RES001,UNIT=2311
//SYSIN DD *
IEF236I ALLOC. FOR DSPLAY   DISVTOC
IEF237I SYSRES   ON 191
IEF237I SYSIN    ON 00C
```

```
IEF285I   SYSOUT                                    SYSOUT
IEF285I   VOL SER NOS=      .
IEF285I                                             KEPT
IEF285I   VOL SER NOS= RES001.
```

Wait-State PSW Codes

1- Display Current PSW - Last 2 Bytes

2- Codes:

IPL Problem -

0001 → 0009 + 000F

Supervisor Problem-

00E2
0F01 → 0F03 - Alarm

SER Problem -

0F05 → 0F0A
0F0D + 0F0E

3- Follow Directions
in O/S Operator's Guide

Operator Commands Via 1052

Cancel
Display
Mount
Reply
Req
Set
Start
Stop
Unload
Vary

Free Form in Typing

Except -

- Only 1 Line
- At Least 1 Blank After Command

CANCEL

CANCEL b JOBNAME[,DUMP]

EXAMPLES-

CANCEL b JOB4 (EOB)
CANCEL b SMITH, DUMP (EOB)

DISPLAY

DISPLAY b JOBNAMES

EXAMPLES-

DISPLAY b JOBNAMES (EOB)
DISPLAY b STATUS (EOB)

MOUNT

MOUNT ß UNITNAME [, VOL = (NL, SERIAL#)]

EXAMPLES -

MOUNT ß 190 (EOB)
MOUNT ß 183, VOL = (NL, TAP737) (EOB)

REPLY

REPLY ß ID, 'TEXT'

EXAMPLES -

REPLY ß 03, 'UP' (EOB)
REPLY ß 00, '193' (EOB)

REQ

REQ␣ ⟶

EXAMPLE-

REQ␣(EOB)

SET

SET ␣DATE=YY.DDD[,CLOCK=HH.MM.SS]
[,Q=(UNITNAME[,F])]
[,PROC=UNITNAME]

EXAMPLES-

SET ␣DATE=67.300 (EOB)

SET ␣DATE=67.300,CLOCK=09.53.20 (EOB)

SET ␣DATE=67.300,CLOCK=18.23.00,PROC=
190 (EOB)

START

START b RDR OR WTR, UNITNAME [,JOBNAME]
[,VOL = SERIAL#]
[,DSN = DATASETNAME]
[,FILESEQ = FILE-SEQUENCE-#]

EXAMPLES-

START b RDR,00C (EOB)
START (EOB)
START b RDR,180,JOBDEMO (EOB)
START (EOB)

STOP

STOP b JOBNAMES

EXAMPLE-

STOP b JOBNAMES (EOB)
STOP b STATUS (EOB)

UNLOAD

UNLOAD b UNITNAME

EXAMPLES-

UNLOAD b 185 (EOB)

UNLOAD b 193 (EOB)

VARY

VARY b UNITNAME, ONLINE OR OFFLINE

EXAMPLES-

VARY b 190, OFFLINE (EOB)

VARY b 184, ONLINE (EOB)

Commands To O/S via System Input Reader

- May Use In Lieu of 1052
- Enter Thru Input Stream - RDR
- // b in Positions 1 → 3
- Command Followed By b

Valid Commands

Display	Mount
Set	Start
Stop	Unload
Vary	

Rules

No continued statements
Use prior to Job-Exec - or Null
Don't intersperse with data
May not take effect
Immediately

PCP O/S Operations

IPL

- Set Addr of IPL Volume - Sysres - in Load Unit Dials of Console
- When IPL Volume Device Is Ready,
- Press Load Key on Console -

Causes

Hardware: System Reset

Hardware: Direct Access Arm To Ready at TRK 0 - CYL 0 of IPLed Device

Hardware: Read of First 24 Bytes From TRK 0 - CYL 0 into Main Storage Addr 0 - 26

Hardware: Bytes 0-7 is Made Current PSW

Program Control Begins

Control of S/360 Given To PSW

Bytes 8-15 and 16-23 are Recognized as 2 CCWs

2 CCWs are Executed,

Causing

A Second Record from TRK0-CYL0 To Read Into Main Storage

This Record Receives CPU Control and-

Reads The Initial-Program-Load Program into Main Storage

The IPL Program now Receives CPU Control

IPL Program Function

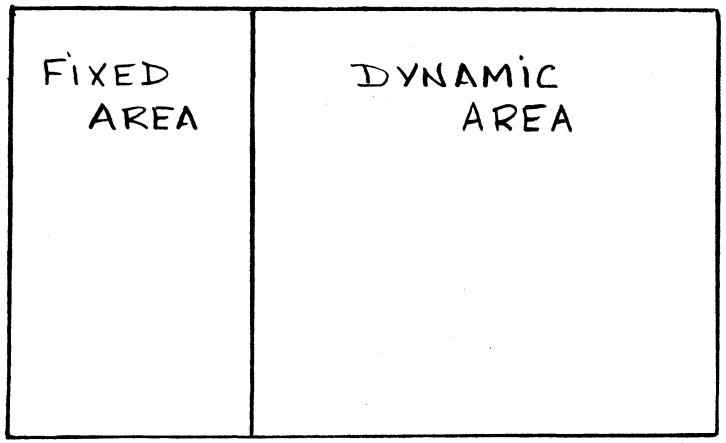
Reads Volume Label of IPL Vol

Finds Addr (TRK + CYL) of VTOC

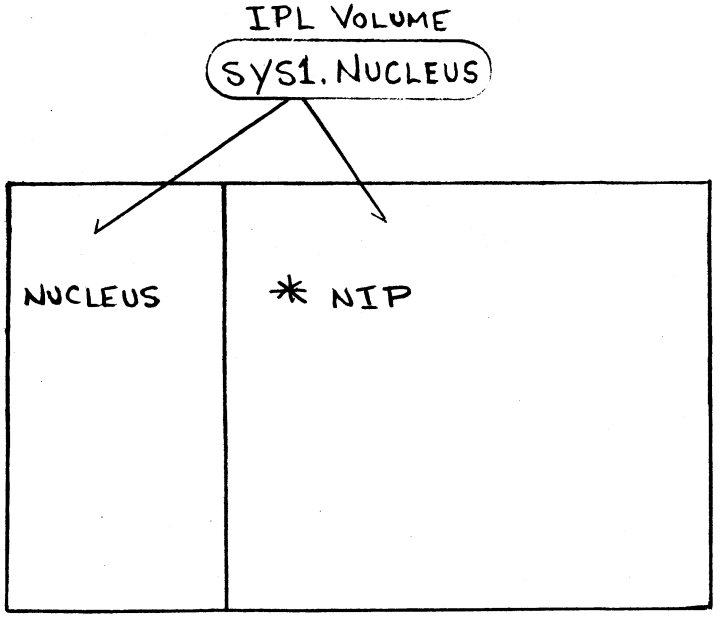
Searches VTOC to find Addr of PDS - SYS1.NUCLEUS

Reads The Resident Portion of the Control Program - Nucleus - into 'Fixed' Area of Main Storage From SYS1.NUCLEUS

Reads The Nucleus Initialization Program - NIP - into the 'Dynamic' Area of Main Storage from SYS1.NUCLEUS



#61



CPU CONTROL GIVEN TO NIP

NIP -

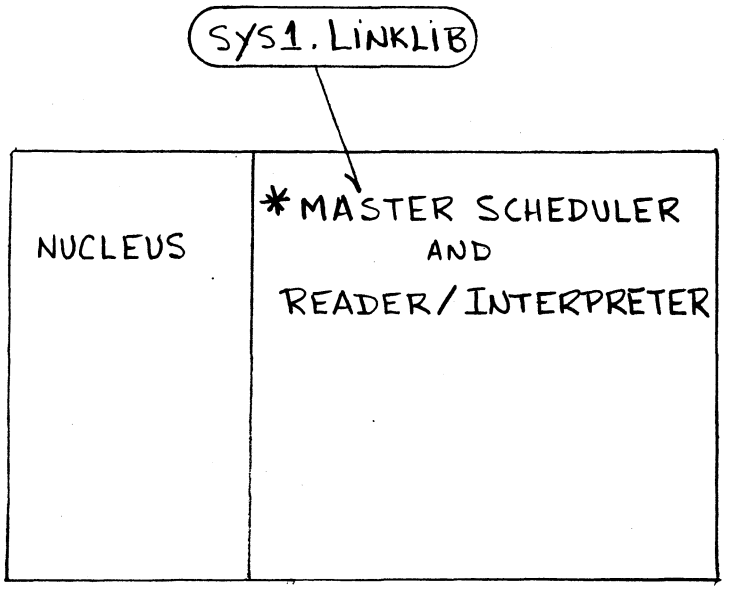
PROCESSES NUCLEUS
DETERMINES SIZE OF CORE
CHECKS STATUS OF TIMER

POSSIBLE 1052 MSG OR
WAIT-STATE PSW CODES
ON ERROR CONDITIONS.

#62

NIP COMPLETES ITS PROCESSING

CONTROL PASSES TO MASTER SCHEDULER



IEE007A READY. — TO OPERATOR

SET DATE ETC. — FROM OPERATOR

IEE013II REJECTED. SET DATE REQ'D.

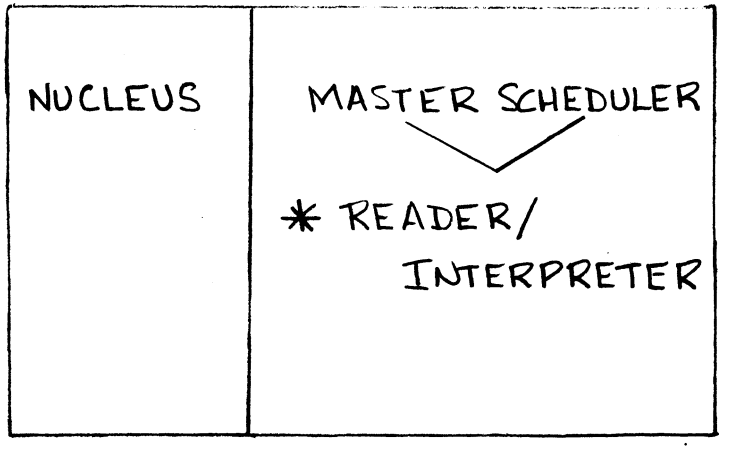
#63

START RDR, xxx
START WTR, xxx > TO OPERATOR

VARY xxx, OFFLINE - FROM OPERATOR
(OPTIONAL)

START

CAUSES CONTROL TO PASS
FROM MASTER SCHEDULER
TO READER/INTERPRETER



#64

READER/INTERPRETER FUNCTIONS

BEGINS READING of JOB CONTROL LANGUAGE STATEMENTS FROM SYSTEM INPUT READER DEVICE - - O/S READER RTN -

READER GIVES INFORMATION (JCL) TO INTERPRETER WHICH TRANSLATES IT INTO FUNCTIONS TO BE PERFORMED.

INTERPRETER'S TRANSLATIONS ARE WRITTEN ON DASD IN THE O/S DATA SET - SYSJOBQE.

JCL

JOB - EXEC - DD - COMMAND
NULL - DELIMITER

GENERAL FORMAT

//NAME␣OPERATION␣OPERAND
FOR JOB - EXEC - DD

//␣OPERATION␣OPERAND
FOR EXEC - DD - COMMAND

// FOR NULL

/* FOR DELIMITER

JOB

Name	Operation	Operand
//jobname	JOB	**Positional Parameters** [([account number][,other accounting information])] [,programmer's name] **Keyword Parameters** [TYPRUN={SETUP / NONSETUP}] [PRTY=job priority] [COND=((code,operator)[,(code,operator)]...)] [MSGLEVEL={0 / 1}] [MSGCLASS=classname]

The Job Statement

JOB: JCL STATEMENTS

```
//JOBXYZ JOB 1,SMITH,MSGLEVEL=1
//FORT   JOB  ,DECKER,MSGLEVEL=0
//PAY    JOB  ,MSGLEVEL=1
//ADS    JOB  1,JONES,TYPRUN=NONSETUP,MSGLEVEL=1
```

#68

1-89 #

EXEC

EXEC: JCL STATEMENTS

Name	Operation	Operand
//[stepname]	EXEC	**Positional Parameters** {PGM=progname PGM=*.stepname.ddname PGM=*.stepname.procstep.ddname PROC=procname procname} **Keyword Parameters** [{TIME TIME.procstep}=(minutes,seconds)] [{COND COND.procstep}= ((code,operator,stepname) [,(code,operator,stepname)]...)] [{PARM PARM.procstep}=(xxxx...)] [{ACCT ACCT.procstep}=(accounting information)]

The Execute Statement

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
//LKED   EXEC PGM=IEWLE180,PARM='XREF,LIST,LET'
//OLDEBC04 EXEC PGM=*.LKED.SYSLMOD
//        EXEC PAYROLL
//STEP1 EXEC PROC=ANALYSIS
```

#69.1

#69

DD

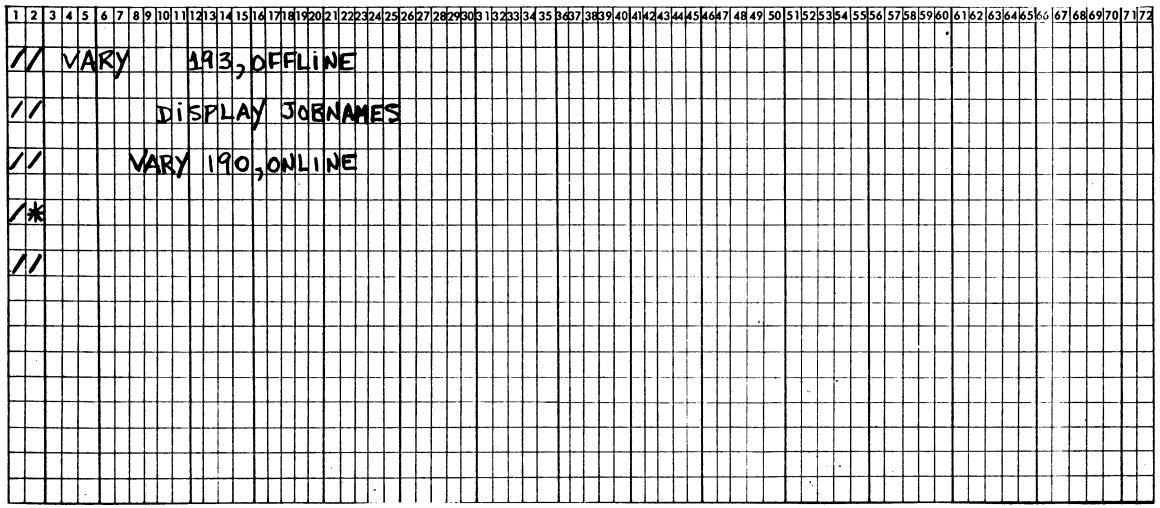
DATA DEFINITION: JCL STATEMENTS

Name	Operation	Operand
//[ddname / poolname / procstep.ddname]	DD	**Positional Parameters** [* / DATA / DUMMY] **Keyword Parameters** [DDNAME=ddname] [DSNAME=(dsname / dsname(element) / *.ddname / *.stepname.ddname / *.stepname.procstep.ddname / &name / &name(element))] [DCB=((dsname / *.ddname / *.stepname.ddname / *.stepname.procstep.ddname), [subparameter list])] [SEP=(subparameter list) / AFF=ddname] [UNIT=(subparameter list)] [SPACE=(subparameter list) / SPLIT=(subparameter list) / SUBALLOC=(subparameter list)] [VOLUME=(subparameter list)] [LABEL=(subparameter list)] [DISP=(subparameter list) / SYSOUT=(subparameter list)]

The Data Definition Statement

```
//SYSUT1 DD   UNIT=2311,SPACE=(TRK,(100,40)),VOLUME=SER=RES4DS,
//             DSNAME=ABC,DISP=(NEW,KEEP)                      X
//OUT    DD UNIT=COD,DSNAME=PUNCH1
//DISKDCB DD  DSNAME=TPDSK,UNIT=2311,DISP=(OLD,KEEP),VOLUME=SER=RES003
//SYSLIN DD   *
//DD1    DD   DSNAME=DATA.SIM,DISP=OLD
//DETAIL DD   DSNAME=XYZ,UNIT=TP7TRK,DISP=(NEW,DELETE),LABEL=(,NL)
//PRINT1 DD   SYSOUT=A
```

#



#72

```
//OLDERCOL JOB 1,TCLARKE,MSGLEVEL=1
//LKED   EXEC  PGM=IEWLE180,PARM='XREF,LIST,LET'
//SYSPRINT DD  SYSOUT=A
//SYSUT1 DD    UNIT=2311,SPACE=(TRK,(100,40)),VOLUME=SER=RESADS,        X
//             DSNAME=&CBX
//SYSLMOD DD   DSNAME=&BAAAT(OLDEBC04),SPACE=(1024,(50,20,1)),          X
//             UNIT=2311,DISP=(NEW,PASS),VOLUME=SER=RESADS
//SYSLIN DD    *
IEF236I ALLOC. FOR OLDERCOL LKED
IEF237I SYSUT1   ON 192
IEF237I SYSLMOD  ON 192
IEF237I SYSLIN   ON 00C
```

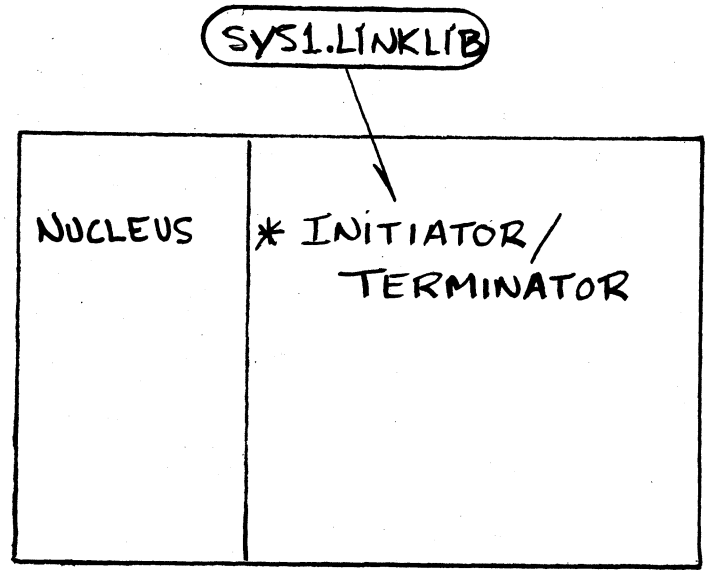
#73

```
//OLDEBC04 EXEC PGM=*.LKED.SYSLMOD
//DISKDCB DD    DSNAME=TPDSK,UNIT=2311,DISP=(OLD,KEEP),VOLUME=SER=RESADS
//DCBTP   DD    DSNAME=TPIO,UNIT=020
//SYSABEND DD UNIT=2311,SPACE=(TRK,(100,10)),VOLUME=SER=RESADS,          C
//             DSNAME=M/VEOUT,DISP=(OLD,KEEP)
//
IEF236I ALLOC. FOR OLDERCOL OLDEBC04
IEF237I PGM=*.DD ON 192
IEF237I DISKDCB  ON 192
IEF237I DCBTP    ON 020
IEF237I SYSABEND ON 192
```

```
//JOB4    JOB   1,HARGREAVES,MSGLEVEL=1
//JOBLIB  DD    DSNAME=PERS.LOAD,DISP=OLD,UNIT=SYSDA,VOLUME=SER=X-C039
//STEP1   EXEC  PGM=TRANEX
//GRAF    DD    UNIT=(2260,1)
//LOG     DD    UNIT=240C,LABEL=(,NL),VOLUME=SER=STOCK,                  C
//              DCB=(RECFM=FB,BLKSIZE=1800,LRECL=100),DISP=(NEW,PASS)
//TERM    DD    DSNAME=TWC,DISP=OLD,UNIT=SYSDA,VOLUME=SER=X-C039
//PREFIL  DD    DSNAME=M03,DISP=OLD,UNIT=SYSDA,VOLUME=SER=X-C039
//SYSPRINT DD   SYSOUT=A
//SYSIN   DD    *
IEF244I   JOB4      STEP1    GRAF    UNABLE TO ALLOCATE FROM AVAILABLE DEVICES
IEF272I - STEP WAS NOT EXECUTED.
```

#74

WHEN R/I COMPLETES PROCESSING



THE I/T IS CALLED INTO CORE
AND RECEIVES CPU CONTROL

#75

Initiator Functions

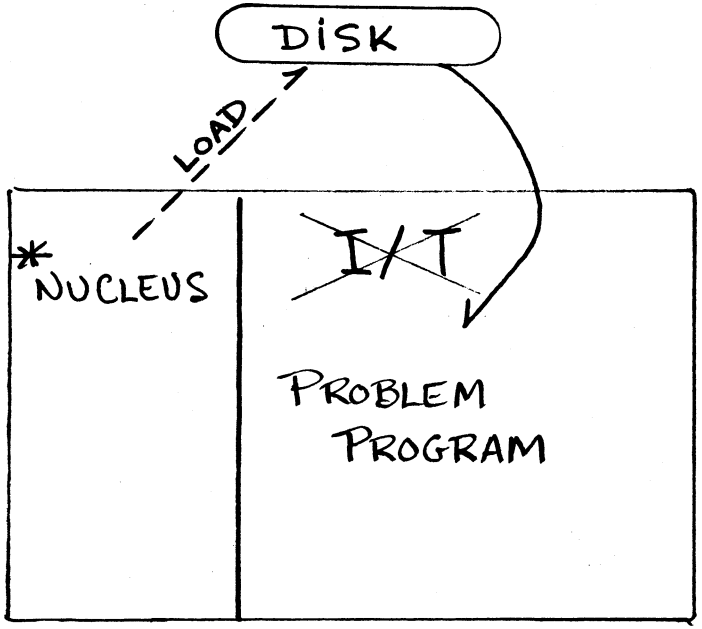
Using Info From Interpreter,
the Initiator:

- Locates Input Data Sets
  by DD Info or
  Catalog Search
- Checks Availability of I/O Devices
- Assigns I/O Devices
- Allocates Aux Stor Space
  by using DADSM Rtn
- Acquires Non-DA Space

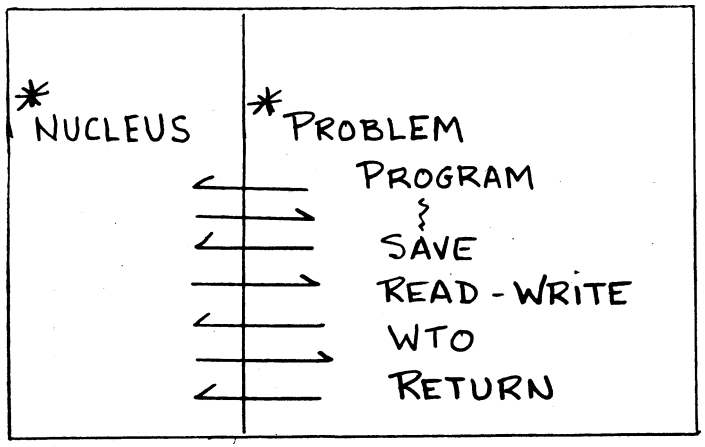
Then -

CPU Control is turned
over to the
NUCLEUS'
Task Mgt Rtns

Supervisor Control



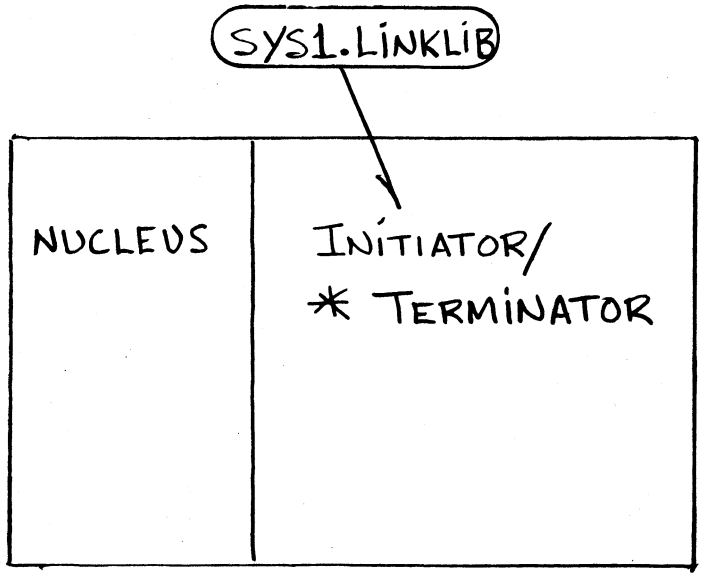
After Loading - The Supervisor
Turns Control over to
the Problem Program



AT EOJ PROBLEM PROGRAM RETURNS CONTROL (FINAL) TO SUPERVISOR

(ABNORMAL EOJ → DUMP)

SUPERVISOR LOADS I/T



CPU CONTROL PASSES TO TERMINATOR

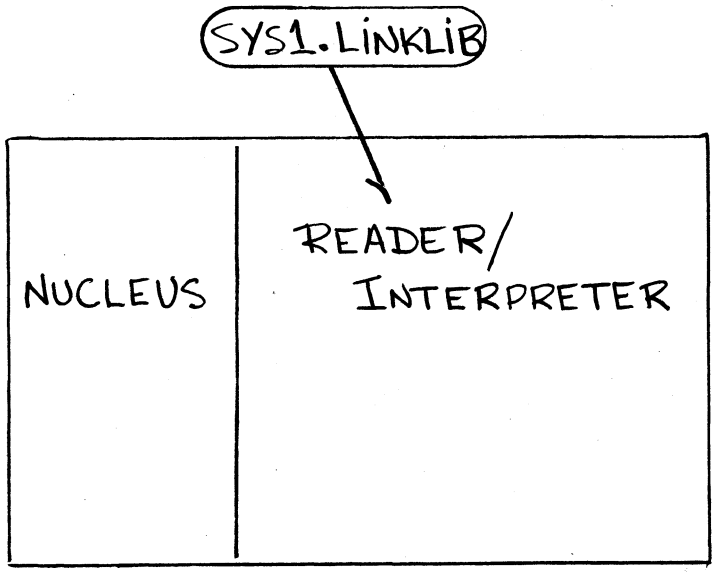
ACCESSES SYSJOBQE
TERMINATES STEP AND/OR JOB
GIVES FINAL DISPOSITION OF DATA SETS USED.

```
//ADSPUNCH EXEC PGM=*.LKED.SYSLMOD
//DISK    DD    DSNAME=TPDSK,UNIT=2311,DISP=(OLD,KEEP),VOLUME=SER=RESADS
//OUT     DD    UNIT=00D,DSNAME=QTQT
//SYSABEND DD   SYSOUT=A
//
IEF236I ALLOC. FOR TESTPCH  ADSPUNCH
IEF237I PGM=*.DD ON 190
IEF237I DISK      ON 193
IEF237I OUT       ON 00D
```

```
IEF285I   BAAAT.TESTPCH                          PASSED
IEF285I   VOL SER NOS= RES002.
IEF285I   TPDSK                                  KEPT
IEF285I   VOL SER NOS= RESADS.
IEF285I   QTQT                                   DELETED
IEF285I   VOL SER NOS=       .
IEF285I   BAAAT.TESTPCH                          DELETED
IEF285I   VOL SER NOS= RES002.
```

#80

TERMINATOR FUNCTION COMPLETED



CYCLE OF
READ - INTERPRET
INITIATE
EXECUTE
TERMINATE
BEGINS AGAIN.

#81