Xerox Control Program-Five (CP-V)

Sigma 6/7/9 Computers

System Management Reference Manual

90 16 74F

Xerox Control Program-Five (CP-V)

Sigma 6/7/9 Computers

System Management Reference Manual

90 16 74F

March 1974

Price: $11.25

REVISION

This publication is a revision of the Control Program-Five (CP-V)/SM Reference Manual, Publication Number 90 16 74E (dated June 1973). This edition documents the B00 version of CP-V. A change in text from that of the previous manual is indicated by a vertical line in the margin of the page.

RELATED PUBLICATIONS

Title	Publication No.
Xerox Control Program-Five (CP-V)/TS Reference Manual	90 09 07
Xerox Control Program-Five (CP-V)/TS User's Guide	90 16 92
Xerox Control Program-Five (CP-V)/OPS Reference Manual	90 16 75
Xerox Control Program-Five (CP-V)/BP Reference Manual	90 17 64
Xerox Control Program-Five (CP-V)/RP Reference Manual	90 30 26
Xerox Control Program-Five (CP-V)/Common Index	90 30 80
Xerox EASY/LN, OPS Reference Manual	90 18 73
Xerox BASIC/Reference Manual	90 15 46
Xerox Extended FORTRAN IV/LN Reference Manual	90 09 56
Xerox Extended FORTRAN IV/OPS Reference Manual	90 11 43
Xerox Extended FORTRAN IV/Library Technical Manual	90 15 24
Xerox FORTRAN Debug Package (FDP)/Reference Manual	90 16 77
Xerox FLAG/Reference Manual	90 16 54
Xerox Meta-Symbol/LN, OPS Reference Manual	90 09 52
Xerox ANS COBOL/LN Reference Manual	90 15 00
Xerox ANS COBOL/OPS Reference Manual	90 15 01
Xerox ANS COBOL/On-Line Debugger Reference Manual	90 30 60
Xerox Manage/Reference Manual	90 16 10
Xerox APL/LN, OPS Reference Manual	90 19 31
Xerox Sort-Merge/Reference Manual	90 11 99
Xerox 1400 Series Simulator/Reference Manual	90 15 02
Xerox Sigma 5/7 Mathematical Routines/Technical Manual	90 09 06
Xerox General Purpose Discrete Simulator (GPDS)/Reference Manual	90 17 58
Xerox Data Management System (DMS)/Reference Manual	90 17 38
Xerox SL-1/Reference Manual	90 16 76
Xerox CIRC-DC/Reference Manual and User's Guide	90 16 97
Xerox CIRC-AC/Reference Manual and User's Guide	90 16 98
Xerox CIRC-TR/Reference Manual and User's Guide	90 17 86

Manual Content Codes: BP - batch processing, LN - language, OPS - operations, RP - remote processing, RT - real-time, SM - system management, TS - time-sharing, UT - utilities.

CONTENTS

APPENDIXES

FIGURES

TABLES

PREFACE

This manual is the principal source of reference information for the system management features of CP-V. It defines the rules for selecting hardware for a CP-V system, generating a CP-V system, authorizing users, maintaining user accounting records, monitoring system performance, and other related functions.

Manuals describing other features of CP-V are outlined below:

- The CP-V Batch Reference Manual, 90 17 64, is the principal source of reference information for the batch processing features of CP-V (i.e., job control commands, system procedures, I/O procedures, program loading and execution, debugging aids, and service processors).

- The CP-V Time-Sharing Reference Manual, 90 09 07, is the principal source of information for the time-sharing features of CP-V. It defines the rules for using the Terminal Executive Language and other terminal processors.

- The CP-V Time-Sharing User's Guide, 90 16 92, describes how to use the various time-sharing features. It presents an introductory subset of the features in a format that allows the user to learn the material by using the features at a terminal as he reads through the document.

- The CP-V Remote Processing Reference Manual, 90 30 26, is the principal source of information about the remote processing features of CP-V. All information about remote processing for all computer personnel (remote and local users, system managers, remote site operators, and central site operators) is included in the manual.

- The CP-V Operations Reference Manual, 90 16 75, is the principal source of reference information for CP-V computer operators. It defines the rules for operator communication (i.e., key-ins and messages), system start-up and initialization, job and system control, peripheral device handling, recovery and file preservation.

- The CP-V Common Index (90 30 80) is an index to all of the above CP-V manuals.

Information for the language and application processors that operate under CP-V is also described in separate manuals. These manuals are listed on the Related Publications page of this manual.

COMMAND SYNTAX NOTATION

Notation conventions used in command specifications and examples throughout this manual are listed below.

Notation	Description
lowercase letters	Lowercase letters identify an element that must be replaced with a user-selected value. CRndd could be entered as CRA03.
CAPITAL LETTERS	Capital letters must be entered as shown for input, and will be printed as shown in output. DPndd means "enter DP followed by the values for ndd".
[ ]	An element inside brackets is optional. Several elements placed one under the other inside a pair of brackets means that the user may select any one or none of those elements. [KEYM] means the term "KEYM" may be entered.
{ }	Elements placed one under the other inside a pair of braces identify a required choice. $\left\{\begin{matrix}\text{A}\\ \text{id}\end{matrix}\right\}$ means that either the letter A or the value of id must be entered.
...	The horizontal ellipsis indicates that a previous bracketed element may be repeated, or that elements have been omitted. name[,name]... means that one or more name values may be entered, with a comma inserted between each name value.
⋮	The vertical ellipsis indicates that commands or instructions have been omitted. MASK2 DATA,2 X'1EF' ⋮ BYTE DATA,3 BA(L(59)) means that there are one or more statements omitted between the two DATA directives.
Numbers and special characters	Numbers that appear on the line (i.e., not subscripts), special symbols, and punctuation marks other than dotted lines, brackets, braces, and underlines appear as shown in output messages and must be entered as shown when input. (value) means that the proper value must be entered enclosed in parentheses; e.g., (234).
Subscripts	Subscripts indicate a first, second, etc., representation of a parameter that has a different value for each occurrence. $sysid_1$,$sysid_2$,$sysid_3$ means that three successive values for sysid should be entered, separated by commas.
Superscripts	Superscripts indicate shift keys to be used in combination with terminal keys. c is control shift, and s is case shift. L^{cs} means press the control and case shift (CONTROL and SHIFT) and the L key.
Underscore	All terminal output is underscored; terminal input is not. $\underline{!}$RUN means that the exclamation point was sent to the terminal, but RUN was typed by the terminal user.
ESC RET LF	These symbols indicate that an ESC (ESC), carriage return (RET), or line feed (LF) character has been sent. $\underline{!}$EDIT RET means that, after typing EDIT, a carriage return character has been sent.

GLOSSARY

ANS tape a tape that has labels written in American National Standard (ANS) format.

auto-call processor a user-specified processor that is automatically connected to the user's terminal when he logs on.

batch job a job that is submitted to the batch job stream through the central site card reader, through an on-line terminal (using the BATCH processor), or through a remote terminal.

binary input input from the device to which the BI (binary input) operational label is assigned.

charge class an arbitrary classification of users to which users and rate tables may be assigned for accounting purposes.

charge rate table a table containing user-assigned charge values for each one of a number of chargeable use accounting resources.

compute time (performance control) the time spent computing during one interaction period.

concatenation a process whereby a number of files with the same filename and format are treated as one logical file. Concatenation is only applicable to ANS tapes.

conflicting reference a reference to a symbolic name that has more than one definition.

control command any control message other than a key-in. A control command may be input via any device to which the system command input function has been assigned (normally a card reader).

control message any message received by the monitor that is either a control command or a control key-in.

control parameter (performance control) a system parameter that can be modified to tune the system.

cooperative a monitor routine that transfers information between a user's program and disk storage (also see "symbiont").

data control block (DCB) a table in the user's program that contains the information used by the monitor in the performance of an I/O operation.

external reference a reference to a declared symbolic name that is not defined within the object module in which the reference occurs. An external reference can be satisfied only if the referenced name is defined by an external load item in another object module.

file extension a convention that is used when certain system output DCBs are opened. Use of this convention causes the file (on RAD, tape, disk pack, etc.) connected to the DCB to be positioned to a point just following the last record in the file. When additional output is produced through the DCB, it is added to the previous contents of the file, thereby extending the file.

function parameter table (FPT) a table through which a user's program communicates with a monitor function (such as an I/O function).

ghost job a job that is neither a batch nor an on-line program. It is initiated and logged on by the monitor, the operator, or another job and consists of a single job step. When the ghost program exits, the ghost is logged off.

global symbol a symbolic name that is defined in one program module and referenced in another.

GO file a temporary disk storage file consisting of relocatable object modules formed by a processor.

granule a block of disk sectors large enough to contain 512 words (a page) of stored information.

interaction time the time between the completion of one terminal input command and the completion of the next.

item (performance control) a control parameter or use item.

job information table (JIT) a table associated with each active job. The table contains accounting, memory mapping, swapping, terminal DCB (M:UC), and temporary monitor information.

job step a subunit of job processing such as compilation, assembly, loading, or execution. Information from certain commands (JOB, LIMIT, and ASSIGN) and all temporary files created during a job step are carried from one job step to the next but the steps are otherwise independent.

key a data item consisting of 1-31 alphanumeric characters that uniquely identifies a record.

key-in information entered by the operator via a keyboard.

library load module a load module that may be combined with relocatable object modules, or other library load modules, to form a new executable load module.

linking loader a program that is capable of linking and loading one or more relocatable object modules and load modules.

load map a listing of loader output showing the location or value of all global symbols entering into the load. Also shown are symbols that are not defined or have multiple definitions.

load module (LM) an executable program formed by the linking loader, using relocatable object modules (ROMs) and/or modules (LMs) as input information.

logical device a peripheral device that is represented in a program by an operational label (e.g., BI or PO) rather than by specific physical device name.

logical device stream an information stream that may be used when performing input from or output to a symbiont device. At SYSGEN, up to 15 logical device streams are defined. Each logical device stream is given a name (e.g., L1, P1, C1), each is assigned to a default physical device, and each is given default attributes. The user may perform I/O through a logical device stream with the default physical device and attributes or he may change the physical device and/or attributes to satisfy the requirements of his job.

monitor a program that supervises the processing, loading, and execution of other programs.

object language the standard binary language in which the output of a processor is expressed.

object module the series of records containing the load information pertaining to a single program or subprogram (i.e., from the beginning to the end). Object modules serve as input to the Load processor or Link processor.

on-line job a job that is submitted through an on-line terminal by a command other than the BATCH command.

operational label a symbolic name used to identify a logical system device.

overlay loader a monitor routine that loads and links elements of overlay programs.

overlay program a segmented program in which the element (i.e., segment) currently being executed may overlay the core storage area occupied by a previously executed element.

physical device a peripheral device that is referred to by a name specifying the device type, I/O channel, and device number (also see "logical device").

program product a compiler or application program that has been or will be released by Xerox, but is not required by all Sigma users and is therefore made available by Xerox on an optional basis. Program products are provided only to those users who execute a License Agreement for each applicable Sigma installation.

prompt character a character that is sent to the terminal by an on-line language processor to indicate that the next line of input may be entered.

protective mode a mode of tape protection in which only ANS expired tapes may be written on through an ANS DCB; no unexpired ANS tape may be written on through a non-ANS DCB; all ANS tapes must be initialized by the Label processor; no tape serial number specification is allowed at the operator's console; specification of an output serial number in an ANS DCB forces processing to be done only on a tape already having that serial number; tapes mounted as IN may not be written; and tapes mounted as other than IN must have a write ring. (See "semiprotective mode".)

public library a set of library routines declared at SYSGEN to be public (i.e., to be used in common by all concurrent users).

reentrant an attribute of a program that allows the program to be shared by several users concurrently. Shared processors in CP-V are may reentrant. That is, each instance of execution of the single copy of the program's instructions has a separately mapped copy of the execution data.

relative allocation allocation of virtual memory to a user program starting with the first unallocated page available.

relocatable object module (ROM) a program or subprogram in Xerox Sigma object language generated by a processor such as Meta-Symbol or FORTRAN.

remote processing an extension of the symbiont system that provides flexible communication between CP-V and a variety of remote terminals.

resident program a program that has been loaded into a dedicated area of core memory.

response time the time between the completion of terminal input and the first program activation.

scheduler a monitor routine that controls the initiation and termination of all jobs, job steps, and time slice quanta.

secondary storage any rapid-access storage medium other than core memory (e.g., RAD storage).

semi-protective mode a mode of tape protection in which a warning is posted to the operator when an ANS DCB attempts output on a non-ANS tape or an unexpired ANS tape, when a non-ANS DCB attempts output on an unexpired ANS tape, or when a tape mounted as INOUT has no write ring. The operator can authorize the overwriting of the tape or the override of INOUT through a key-in (OVER and READ). ANS tapes may be initialized by the Label processor or may be given labels as the result of an operator key-in; tape serial number specification is allowed at the operator's console; and specification of an output serial number in an ANS DCB forces processing to be done only on a tape already having that serial number unless the operator authorizes an overwrite. (See "protective mode".)

session time   the time between terminal log-on and log-off.

shared processor   a program (e.g., FORTRAN) that is shared by all concurrent users. Shared processors must be established during SYSGEN or via DRSP.

siding (performance control)   the text that appears to the left of a value in a performance display to help explain the value.

source language   a language used to prepare a source program suitable for processing by an assembler or compiler.

special shared processor   a shared processor that may be in core memory concurrently with the user's program (e.g., Delta, TEL, or the FORTRAN library).

specific allocation   allocation of a specific page of unallocated virtual memory to a user program.

SR1, SR2, SR3, and SR4   see "system register", below.

static core module   a program module that is in core memory but is not being executed.

stream-id   the name of a logical device stream.

symbiont   a monitor routine that transfers information between disk storage and a peripheral device independent of and concurrent with job processing.

symbolic input   input from the device to which the SI (symbolic input) operational label is assigned.

symbolic name   an identifier that is associated with some particular source program statement or item so that symbolic references may be made to it even though its value may be subject to redefinition.

SYSGEN   see "system generation", below.

system generation (SYSGEN)   the process of creating an operating system that is tailored to the specific requirements of an installation. The major SYSGEN steps include: gathering the relevant programs, generating specific monitor tables, loading monitor and system processors, and writing a bootable system tape.

system library   a group of standard routines in object-language format, any of which may be incorporated in a program being formed.

system register   a register used by the monitor to communicate information that may be of use to the user program (e.g., error codes). System registers SR1, SR2, SR3, and SR4 are current general registers 8, 9, 10, and 11, respectively.

task control block (TCB)   a table of program control information built by the loader when a load module is formed. The TCB is part of the load module and contains the data required to allow reentry of library routines during program execution or to allow entry to the program in cases of traps, breaks, etc. The TCB is program associated and not task associated.

task turnaround time   see "user response time".

thinking and typing time   the time between the terminal read that is issued by the program and the end of the user response (input complete).

tuning a system   the modification of an operating system to adjust system resources to meet changing requirements.

unsatisfied reference   a symbolic name that has been referenced but not defined.

use distribution   a performance distribution that shows the percentage of occurrences of a particular kind of event that fall within a given range on an appropriate scale.

use group   a group of related performance use values and text in the form of siding and headers to explain the values.

use item   a performance use distribution, use group, or siding.

user response time   the time from the completion of the input command until the first character of output is produced, or the next terminal read if no output occurs. This time includes system response, queue delays due to other users, and the processing time of the user's program or processor.

1. INTRODUCTION

CP-V SERVICES

Control Program-Five (CP-V) is a comprehensive operating system designed for use with Sigma 6/7/9 computers and a variety of peripheral equipment. The current release of CP-V offers

- On-line time-sharing, batch processing, remote processing, and real-time services.
- Ability to handle a large number of concurrent users.
- High efficiency due to hardware relocation map, shared reentrant processors, multiple I/O processors, and device pooling.
- A complete recovery system coupled with preservation of user files to provide fast restart following hardware malfunction.
- For on-line users: highly efficient and extensive software, file saving feature, fast response time.
- For batch users: on-line entry, local and remote entry to an efficient multiprogramming batch job scheduler.
- For installation managers: thorough system monitoring and reporting, control and tuning ability, extensive error checking and recovery features.
- For all users: comprehensive accounting and a complete set of powerful processors.

An additional mode of operation, transaction processing, will be incorporated into the system in the near future.

TIME-SHARING AND BATCH PROCESSING

CP-V allows multiple on-line terminal users to concurrently create, debug, and execute programs. Concurrent to time-sharing, CP-V allows up to 16 batch processing jobs to execute in its multiprogramming environment. An efficient multi-batch scheduler selects batch jobs for execution according to priority, job requirements, and availability of resources. Batch jobs may be submitted to this scheduler from a local batch entry device such as a card reader, from an on-line user's terminal, or from a remote site such as a remote batch terminal.

Time-sharing and batch users have access to a variety of powerful and comprehensive language processors and facilities. These processors and facilities are listed below.

Processor	Function
TEL	Executive language control of all terminal activities. (On-line only.)
EASY	Creation, manipulation, and execution of FORTRAN and BASIC programs and data files. (On-line only.)
Edit	Composition and modification of programs and other bodies of text. (On-line only.)
FORTRAN IV	Compilation of Extended FORTRAN IV programs.
COBOL	Compilation of ANS COBOL programs.
Meta-Symbol	Assembly of high-level assembly language programs.
BASIC	Compilation and execution of programs or direct statements written in an extended BASIC language.
APL	Interpretation and execution of programs written in the APL language.
FLAG	Compilation of fast "load-and-go" FORTRAN programs.
FDP	Dubugging of Extended FORTRAN IV programs.
Delta	Debugging of programs at the assembly language level. (On-line only.)
COBOL On-line Debugger	Debugging of ANS COBOL programs. (On-line only.)
PCL	Transfer (and conversion) of data between peripheral devices.
Link	Linkage of programs for execution.
Load	Linkage of programs for execution. (Batch only.)
Batch	Submission of batch jobs via an on-line terminal or another batch job.
Manage	File retrieval, updating, and reporting.
SL-1	Compilation of programs written in a language designed specifically for digital or hybrid simulation.
CIRC	Analysis of electronic circuits.

Processor	Function
EDMS	Organization, storing, updating, and deletion of information in a centralized data base.
Sort/Merge	Sorting and/or merging of records in one or more files.
GPDS	Experimentation with and evaluation of system methods, processes, and designs. (Batch only.)
1400 Series Simulators	Simulation of 1400 series computers. (Batch only.)

REMOTE PROCESSING

The remote processing system is an extension of the CP-V symbiont system. Its purpose is to provide for flexible communication between CP-V and a variety of remote terminals. These terminals can range from a simple card reader, card punch, and line printer combination to another computer system with a wide variety of peripheral devices. Any CP-V user (batch, on-line, ghost) can communicate with any number of devices at one or several remote sites. Because CP-V can act as a central site to some remote sites and simultaneously as a remote terminal to other computers, the remote processing facilities encourage the construction of communication networks.

REAL-TIME PROCESSING

The real-time services provided by CP-V allow users to connect interrupts to mapped programs, control the state of interrupts (e.g., trigger, arm/disarm, enable/disable), clear interrupts either at the time of occurrence or upon completion of processing, and disconnect interrupts no longer required. Users may also request that a mapped program be held in core in order to reduce the time required to respond to an external event (via an interrupt) or to allow various forms of special I/O to occur. Programs may be connected to one of the monitor's clocks such that after a specified period of time, a specified routine is entered. In addition, dedicated foreground memory may be used as inter-program communication buffers or as dedicated memory for unmapped, master mode programs which may be directly connected to external interrupts or real-time clocks.

SYSTEM MANAGEMENT FACILITIES

The manager of each CP-V installation must evaluate his performance requirements before he can effectively use the system management facilities. This evaluation must take place prior to equipment selection since an effective equipment selection can be made only with complete knowledge of the intended use of CP-V.

The performance requirements that must be defined include such things as the portions of system resources that must be devoted to each class of service and the type of service desired. In defining the type of service desired, both the batch turnaround time that is acceptable and the interactive delays that are tolerable must be defined. Other information that will affect system performance includes the number of on-line users to be allowed, the maximum core memory to be allowed each user, and the maximum file space to be allowed each on-line user.

Once an effective selection of equipment has been made and the system has been installed, the system manager may exercise control over the performance of the system through several facilities. These facilities include

- System Generation
- Performance Monitoring and Control
- File Backup Control
- Log-on Supervisory Control
- Use Accounting
- Operations Control

At the time a system is generated, a number of parameters may be defined to tailor the system to the specific requirements of the installation. These parameters include such things as:

1. Maximum core size allowed on-line users.
2. Maximum number of on-line users.
3. Number of characters at which to block terminal output.
4. Number of characters at which to unblock terminal output.
5. Maximum number of tape drives and disk spindles allowed users.
6. Amount of uninterrupted compute time guaranteed a user after selection.
7. Size of time slice quanta.
8. Cutoff limits for peripheral output by batch and on-line users.

After the system has been generated and put into operation, the system manager may dynamically control the performance of the system through the use of a control program. This program is specifically designed to

1. Measure how well the system is performing.

2. Warn of immediate problems (e.g., permanent storage is filling up, response time is becoming noticeably slower, large numbers of terminal errors are occurring).

3. Help "tune" the system for both current and general conditions.

4. Measure the importance of various parts of the system such as the relative use of various processors in terms of CPU time (this might have implications in determining whether a particular processor is dropped or whether its use justifies the effort to add new capabilities).

In addition, the system manager may control the level of backup provided for user files by setting the frequency of operation of the automatic file backup facility. The automatic file backup facility periodically writes files that are stored on disk storage onto magnetic tape so the files can be restored in case of system failure.

The system manager and the computer operator control the amount of disk storage space that is available to the users. Any time the number of disk granules available for file storage falls below a threshold level specified by the system manager, all expired files are automatically deleted and the operator is advised of the number of granules available for files both before and after the expired files are deleted. The operator may change the threshold level whenever necessary. In addition, the operator may initiate a file purge that deletes all files that have not been accessed since a specified date. He may also initiate a file purge that deletes sufficient files, in order of recency of access (from least recent to most recent), to bring the number of available granules up to a specified level.

Another system management facility is the log-on control feature. This feature gives the system manager the means of adding or deleting users and of controlling the privileges granted to users, such as core size and the use of I/O devices. It also allows the system manager to specify a processor other than the Terminal Executive Language to which an on-line user will be automatically connected when he logs on.

CP-V has an extensive user accounting system. Statistics maintained for each account number include

1. CPU time used.
2. Number of file I/O service calls.
3. Number of terminal interactions.
4. Terminal connect time.
5. Number of tape reels or disk packs mounted.
6. File storage used.
7. Number of cards read.
8. Number of pages printed.
9. Core storage used.

These statistics are automatically multiplied by charge rates in rate tables that are defined by the system manager. Accounting charges are listed at the end of every job, and a subset of the statistics and charges is listed when an on-line user logs off. Current values of statistics may be listed by an on-line user through the use of a terminal command.

CP-V has several operational control features that allow the system manager to exercise control over operations through the computer operator. The computer operator may

1. Error and abort users.
2. Send messages to users.
3. Shut down and start on-line and remote batch services.
4. Control symbionts.
5. Respond to hardware errors.
6. Control mounting and dismounting of magnetic tapes and disk packs.

These functions are carried out through a console that also provides a log of overall system operation.

Thus, within reasonable limits, the system may be modified by system management facilities to meet changing performance requirements. Beyond these limits, control must be exercised by direct management fiat and by education of users.

2. SYSTEM OVERVIEW

INTRODUCTION

The CP-V operating system consists of a monitor and a number of associated processors (Figure 1). The monitor provides overall supervision of program processing. The associated processors provide specific functions such as compilation, execution, and debugging.

PROCESSORS

Processors are illustrated in Figure 1 at two levels. The upper level contains executive language and related processors. The lower level contains all other processors. These processors are defined in the following paragraphs.

COMMAND PROCESSORS

The four processors in this group are: LOGON/LOGOFF, EASY, TEL, and CCI. The first of these processors is available to on-line and batch users, the second and third are available to on-line users only, and the last is available to batch users only.

LOGON/LOGOFF

LOGON admits on-line users to the system and connects the user's terminal either to TEL or to an alternative processor, such as BASIC, that has been selected by the user. LOGOFF disconnects a user from the system and does the final cleanup and accounting.

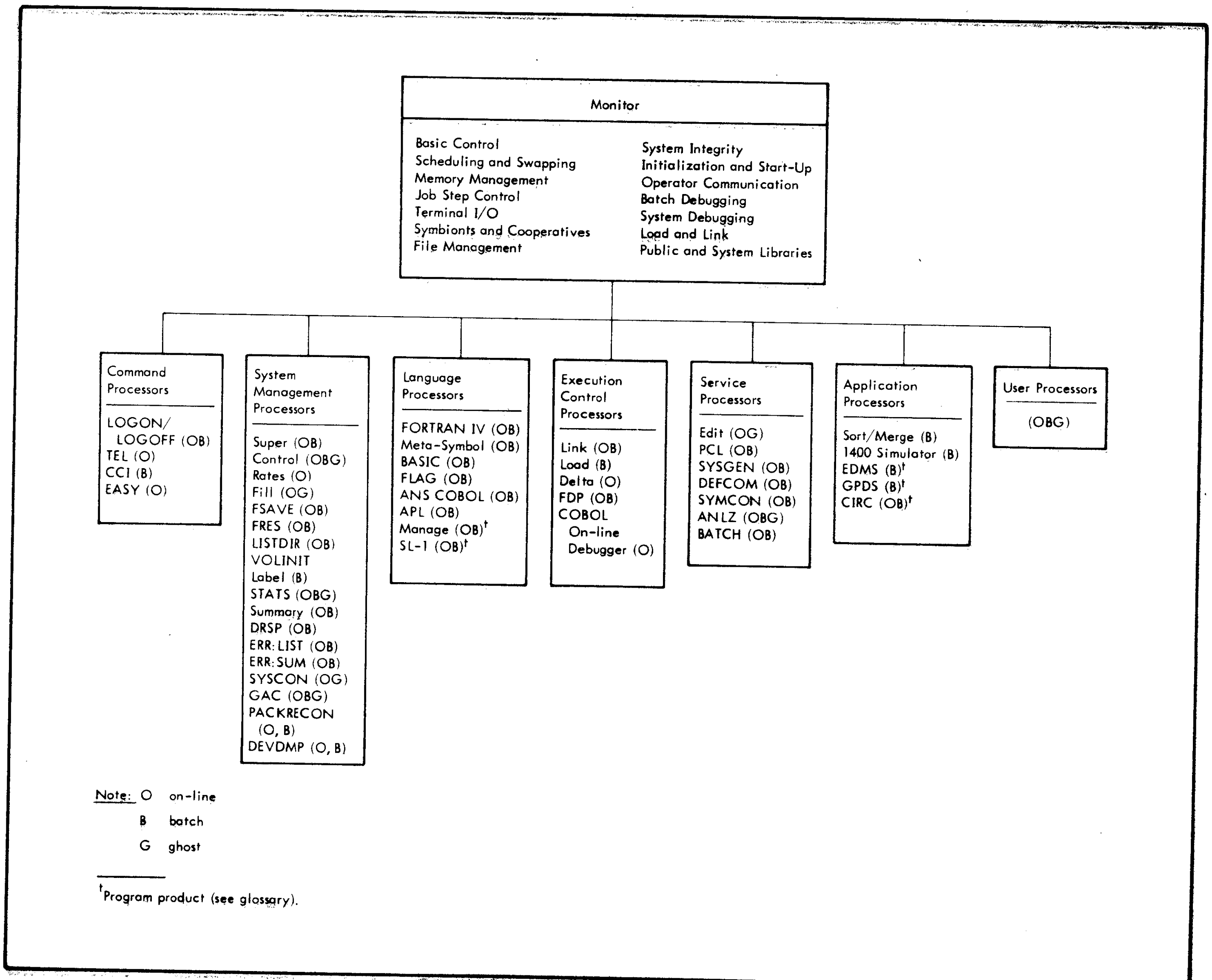


Figure 1. CP-V Operating System

EASY

EASY is a shared processor that enables the user to create, edit, execute, save, and delete program files written in BASIC or FORTRAN. EASY also allows the user to create and manipulate EBCDIC data files. Although intended primarily for Teletype® operations, EASY can be used with any type of on-line terminal supported by the system. (Reference: EASY/LN, OPS Reference Manual, 90 18 73.)

TERMINAL EXECUTIVE LANGUAGE

The Terminal Executive Language (TEL) is the principal terminal language for CP-V. Most activities associated with FORTRAN, COBOL and assembly language programming can be carried out directly in TEL. These activities include such major operations as composing programs and other bodies of text, compiling and assembling programs, linking object programs, initiating execution, and debugging programs. They also include such minor operations as saving and restoring core images of programs for which execution was interrupted, determining program status, and setting simulated tab stops. (Reference: CP-V/TS Reference Manual, 90 09 07.)

CONTROL COMMAND INTERPRETER

The Control Command Interpreter is the batch counterpart of TEL. It provides the batch user with control over the processing of batch programs just as TEL provides on-line users with control over the processing of on-line programs. (Reference: CP-V/BP Reference Manual, 90 17 64.)

SYSTEM MANAGEMENT PROCESSORS

System management processors furnish the manager of a CP-V installation with on-line control of the system. Eighteen system management processors are supplied.

SUPER

Super gives the system manager control over the entry of users and the privileges extended to users. Through the use of Super commands, the system manager may add and delete users, specify how much core and disk storage space a user will have, and specify how many central site magnetic tape units a user will have. He may also grant certain users, such as system programmers, special privileges, e.g., the privilege of examining, accessing, and changing the monitor. (Reference: Chapter 4.)

CONTROL

The Control processor provides control over system performance. CP-V has a number of performance measurements built directly into the system. Commands of the Control processor enable the system manager to display these measurements and to "tune" the system as needed by setting new values for the parameters that control system performance. (Reference: Chapter 6.)

RATES

The Rates processor allows the system manager to set relative charge weights on the utilization of system services.

Specific items to which charge weights may be assigned include

1. CPU time.
2. CPU time multiplied by core size.
3. Terminal interactions.
4. I/O CALs.
5. Console minutes.
6. Tapes and packs mounted.
7. Page-date storage.
8. Peripheral I/O cards plus pages.

(Reference: Chapter 5.)

FILL

The FILL processor performs three basic file maintenance functions:

1. It copies files from disk to tape as a backup.
2. It restores files from tape to disk.
3. It deletes files from disk.

(Reference: CP-V/OPS Reference Manual, 90 16 75.)

®Registered trademark of the Teletype Corporation

FSAVE

The Fast Save (FSAVE) processor is designed to save disk files on tape at or near tape speed. The processor is faster than any other file saving procedure under CP-V. (Reference: CP-V/OPS Reference Manual, 90 16 75.)

FRES

The File Restore (FRES) processor is designed to restore to disk files that were saved on tape by FSAVE. (Reference: CP-V/OPS Reference Manual, 90 16 75.)

LISTDIR

The LISTDIR processor provides a method for obtaining a listing of the number of granules in the file directory, in the file information table, and in the free sector pool of each account. It is also used to release null file directories and the granules in their free sector pools back to the system. (Reference: CP-V/OPS Reference Manual, 90 16 75.)

VOLINIT

VOLINIT provides for the initialization of public and private disk packs. It is used to establish serial numbers and ownership, to write headers and other system information in selected areas of the volumes, and to test the surface of the disks and select alternate tracks to be used in place of flawed tracks. (Reference: CP-V/OPS Reference Manual, 90 16 75.)

LABEL

The Label processor initializes ANS tapes by writing ANS formatted labels. It may also be used to create "unlabeled" tapes from new tapes to be used as scratch tapes and to print the contents of the header and trailer labels of labeled tapes or the first 80 bytes of each block on unlabeled tapes. (Reference: CP-V/OPS Reference Manual, 90 16 75.)

STATS

The STATS processor displays and collects performance data on a running system and produces snapshot files to be displayed by the report generator Summary. (Reference: Chapter 6.)

SUMMARY

The Summary processor provides a global view of system performance by formatting and displaying the statistical data collected by STATS. (Reference: Chapter 6.)

DRSP

DRSP (Dynamic Replacement of Shared Processors) enables the system manager to dynamically add, replace, or delete processors and monitor overlays. He may do this during normal system operations with other users in the system. (Reference: Chapter 10.)

ERR:LIST

The ERR:LIST program examines the hardware error log file ERRFILE for malfunction records that were written during a specified time period and produces a formatted listing of these records with (optionally) a summary of the records for that period. The formatted listing is complete with headings and formatting necessary for easy reading and use by field personnel. (Reference: Chapter 12.)

ERR:SUM

The ERR:SUM program creates a listing that summarizes the hardware errors that have occurred. ERR:SUM accesses the hardware error summary file SUMFILE, a file that is updated automatically by the ghost program ERR:FIL. (Reference: Chapter 12.)

SYSCON

SYSCON is a system control processor that can be used to partition resources from the system, to return resources to the system, and to display the status of the various system resources. SYSCON can also be used to build, update, or display the M:MODNUM file, a file which contains device and controller model numbers. (Reference: Chapter 13.)

GRANULE ACCOUNTING CLEANUP PROCESSOR (GAC)

The Granule Accounting Cleanup (GAC) processor correlates information between the file DISKPOOL and the account authorization file, :USERS. DISKPOOL is created by the FSAVE processor and contains specific account information. Each account record in DISKPOOL contains an entry for accumulated public disk pack granules and an entry for accumulated RAD granules. When GAC is run, these accumulated values are compared against the maximum values for the corresponding accounts in the :USERS file and the user's entry in the :USERS file is updated to reflect the latest accumulated values for RAD and disk. When the accumulated RAD or disk granules exceed the corresponding maximum values, this fact is noted in the report that is produced by the GAC processor. (Reference: CP-V/OPS Reference Manual, 90 16 75.)

PACKRECON

The PACKRECON processor rebuilds the private volume allocation tables on a disk pack when a crash has occurred which affected the disk pack. (Reference: CP-V/OPS Reference Manual, 90 16 75.)

DEVDMP

The Device Save/Restore processor (DEVDMP) is a stand-alone utility program designed to dump entire disk volumes to magnetic tapes for restoration at a later time. Restoration may only be made to an identical storage unit. (Reference: CP-V/OPS Reference Manual, 90 16 75.)

LANGUAGE PROCESSORS

Language processors translate high-level source code into machine object code. Eight processors of special importance are described below. All of these can be used in both on-line and batch mode.

XEROX EXTENDED FORTRAN IV

The Xerox Extended FORTRAN IV language processor consists of a comprehensive algebraic programming language, a compiler, and a large library of subroutines. The language is a superset of most available FORTRAN languages, containing many extended language features to facilitate program development and checkout. The compiler is designed to produce very efficient object code, thus reducing execution time and core requirements, and to generate extensive diagnostics to reduce debugging time. The library contains over 235 subprograms and is available in a reentrant version. Both the compiler and run-time library are reentrant programs that are shared among all concurrent users to reduce the utilization of critical core resources.

The principal features of Xerox Extended FORTRAN IV are as follows:

- Extended language features to reduce programming effort and increase range of applications.
- Extensive meaningful diagnostics to minimize debugging time.
- In-line symbolic code to reduce execution time of critical parts of the program.
- Overlay organization for minimal core memory utilization.
- Compiler produced reentrant programs.

(Reference: Extended FORTRAN IV/LN Reference Manual, 90 09 56, and Extended FORTRAN IV/OPS Reference Manual, 90 11 43.)

META-SYMBOL

Meta-Symbol is a procedure-oriented macro assembler. It has services that are available only in sophisticated macro assemblers and a number of special features designed to permit the user to exercise dynamic control over the parametric environment of assembly. It provides users with a highly flexible language with which to make full use of the available Sigma hardware capabilities.

Meta-Symbol may be used in either batch or on-line mode. When used in on-line mode, the assembler allows programs to be assembled and executed on-line but does not allow conversational interaction.

One of the many Meta-Symbol features is a highly flexible list definition and manipulation capability. In Meta-Symbol, lists and list elements may be conveniently redefined, thus changing the value of a given element.

Another Meta-Symbol feature is the macro capability. Xerox uses the term "procedure" to emphasize the highly sophisticated and flexible nature of its macro capability. Procedures are assembly-time subroutines that provide the user with an extensive function capability. Procedure definition, references, and recursions may be nested up to 32 levels.

Meta-Symbol has an extensive set of operators to facilitate the use of logical and arithmetic expressions. These operators facilitate the parametric coding capabilities available with Meta-Symbol (parametric programming allows for dynamic specification of both "if" and "how" a given statement or set of statements is to be assembled).

Meta-Symbol users are provided with an extensive set of directives. These directives, which are commands intrinsic to the assembly, fall into three classes:

1. Directives that involve manipulation of symbols and are not conditionally executed.

2. Directives that allow parametric programming.

3. Directives that do not allow parametric programming.

A number of intrinsic functions are also included in Meta-Symbol. These give the user the ability to obtain information on both the structure and content of an assembly time construct. For example, the user can acquire information on the length of a certain list. He can inquire about a specific symbol and whether it occurs in a procedure reference. (Reference: Meta-Symbol/LN, OPS Reference Manual, 90 09 52.)

BASIC

BASIC is a compiler and programming language based on Dartmouth BASIC. It is, by design, easy to teach, learn, and use. It allows individuals with little or no programming experience to create, debug, and execute programs via an on-line terminal. Such programs are usually small to medium size applications of a computational nature.

BASIC is designed primarily for on-line program development and execution, or on-line development and batch execution. In addition, programs may be developed and executed in batch mode.

BASIC provides two user modes of operation. The editing mode is used for creating and modifying programs. The compilation/execution mode is used for running completed programs. This arrangement simplifies and speeds up the program development cycle.

Statements may be entered via a terminal and immediately executed. The principal benefit of direct execution is on-line development of programs and short simple computations. During execution, programs may be investigated for loop detection, snapshots of variables may be obtained, values of variables may be changed, flow of execution may be re-routed, and so on. This unique capability allows an on-line terminal to be used as a "super" desk calculator.

At compile and execute time, the user may specify if an array dimension check is to be made. In the safe mode, statements are checked to verify that they do not reference an array beyond its dimensions. In the fast mode, this time consuming check is not made. Thus, the safe mode could be used during checkout, and the fast mode could be used to speed up execution when the program reaches the production stage.

BASIC provides an image statement that uses a "picture" of the desired output format to perform editing. It also has TAB capability and a precision option to indicate the number of significant digits (6 or 16) to be printed.

An easy-to-use feature is provided to allow the user to read, write, and compare variable alphanumeric data. This is particularly important for conversational input processing.

Chaining permits one BASIC program to call upon another for compilation and execution without user intervention. Thus, programs that would exceed user core space may be segmented, and overlay techniques may be employed via the chaining facility. (Reference: BASIC/Reference Manual, 90 15 46.)

FLAG

FLAG (FORTRAN Load and Go) is an in-core FORTRAN compiler that is compatible with the FORTRAN IV-H class of compilers. It can be used in preference to the other FORTRAN compilers when users are in the debugging phase of program development. FLAG is a one-pass compiler and uses the Extended FORTRAN IV library. Included in the basic external functions are the Boolean functions IAND (AND), IEOR (exclusive OR), and IOR (OR), which give the FORTRAN user a bit manipulation capability.

If several FLAG jobs are to be run sequentially, they may be run in a sub-job mode, thus saving processing time normally needed for the Control Command Interpreter (CCI) to interpret the associated control cards. In this mode, FLAG will successively compile and execute any number of separate programs, thereby reducing monitor overhead.

The FLAG debug mode is a user-selected option that generates extra instructions in the compiled program to enable the user, during program execution, to detect errors in program logic that might otherwise go undetected or cause unexplainable program failure. (Reference: FLAG/Reference Manual, 90 16 54.)

ANS COBOL

The Xerox ANS COBOL compiler offers the user a powerful and convenient programming language facility for the implementation of business or commercial applications. The language specifications fully conform to the proposed ANSI standard for the various functional processing modules. Only those language elements that cause ambiguities or are seldom used have been deleted. The compiler's design takes full advantage of Sigma's unique hardware features, resulting in rapid compilation of source code, rapid execution of the resulting object code, and the generation of compact programs.

The result is a highly efficient programming system requiring a minimum amount of storage.

Xerox ANS COBOL contains many facilities that are either not found in other systems or, if available, are provided only at greater cost in terms of equipment required. Some of the facilities that provide more flexibility and ease of use in program development include

1. Implementation of table handling mode.
2. Sort/merge linkage.
3. Sequential access.
4. Random access linkage.
5. Segmentation.
6. Report writer.
7. Library utilization.
8. Calling sequence for FORTRAN, Meta-Symbol, etc.
9. Packed decimal as well as floating-point arithmetic formats.
10. Data name series options for ADD, SUBTRACT, MULTIPLY, DIVIDE, and COMPUTE verbs.

The system provides the user with a comprehensive set of aids to minimize the time required to print "bug-free" programs in the form of listings. These listings include

1. The source language input to the compiler with interspersed English language diagnostic messages.
2. An optional listing of the relocatable binary output, printed in line number sequence identical to the source language listing.
3. A cross-reference listing, indicating by line number where each data name or paragraph name is defined in the COBOL program and where each reference is located.

In addition, at run time, the user may use TRACE and EXHIBIT to follow execution of the procedure division.

The compiler is designed to take full advantage of high-speed, random access secondary storage (e.g., RAD storage). This feature means faster job execution because of minimized I/O delays, and smaller core memory requirements because of rapid overlay service. (Reference: ANS COBOL/LN Reference Manual, 90 15 00.)

APL

APL is an acronym for A Programming Language, the language invented by Kenneth Iverson. It is an interpretive, problem-solving language. As an interpretive language, APL does not wait until a program is completed to compile it into object code and execute it; instead, APL interprets each line of input as it is entered to produce code that is immediately executed. As a problem-solving language, APL requires minimal computer programming knowledge; a problem is entered into the computer and an answer is received, all in the APL language.

Because APL is powerful, concise, easy to learn, and easy to use, it is widely used by universities, engineers, and statisticians. It also has features that make it attractive for business applications where user interaction and rapid feedback are key issues. One of APL's major strengths is its ability to manipulate vectors and multidimensional arrays as easily as it does scalar values. For example, a matrix addition that might require a number of statements and several loops in other languages can be accomplished as A+B in APL. This type of simplification exemplifies APL's concise power. (Reference: APL/LN, OPS Reference Manual, 90 19 31.)

MANAGE (PROGRAM PRODUCT)[†]

Manage is a generalized file management system. It is designed to allow decision makers to make use of the computer to generate and update files, retrieve useful data, and generate reports without having a knowledge of programming.

Manage consists of four subprograms: Dictionary, Fileup, Retrieve, and Report. The Dictionary subprogram is a data file and is the central control element in the Manage system. It consists of definitions and control and formatting parameters that precisely describe the characteristics of a data file. The Fileup subprogram initially creates and then maintains a data file. The Retrieve subprogram extracts data from a data base file according to user-specified criteria. The Report subprogram automatically prepares printed reports from data extracted by the Manage retrieval program. (Reference: Manage/Reference Manual, 90 16 10.)

SIMULATED LANGUAGE (PROGRAM PRODUCT)[†]

The Simulation Language (SL-1) is a simplified, problem-oriented digital programming language designed specifically for digital or hybrid simulation. SL-1 is a superset of CSSL (Continuous System Simulation Language), the standard language specified by Simulation Councils, Inc., for simulation of continuous systems. It exceeds the capabilities of CSSL and other existing simulation languages by providing hybrid and real-time features, interactive debugging features, and a powerful set of conditional translation features.

SL-1 is primarily useful in solving differential equations, a fundamental procedure in the simulation of parallel, continuous systems. To perform this function, SL-1 includes

[†]See "program product" in glossary.

six integration methods and the control logic for their use. In hybrid operations, SL-1 automatically synchronizes the problem solution to real-time and provides for hybrid input and output.

Because of the versatility of Xerox Sigma computing systems and the broad applicability of digital and hybrid simulation techniques, applications for SL-1 exist across the real-time spectrum. The library concept of SL-1 allows the user to expand upon the Xerox supplied macro set and facilitates the development of macro libraries oriented to any desired application. (Reference: SL-1/Reference Manual, 90 16 76.)

EXECUTION CONTROL PROCESSORS

Processors in this group control the execution of object programs. Delta and COBOL On-Line Debugger can be used in on-line mode only. Load can be used in batch mode only. Link and FDP can be used in either batch or on-line mode.

LINK

Link is a one-pass linking loader that constructs a single entity called a load module, which is an executable program formed from relocatable object modules (ROMs). Link is designed to make full use of mapping hardware. It is not an overlay loader. If the need for an overlay loader exists, the overlay loader (Load) must be called and the job must be entered in the batch stream. (Reference: CP-V/TS Reference Manual, 90 09 07.)

LOAD

Load is a two-pass overlay loader. The first pass processes

1. All relocatable object modules (ROMs).

2. Protection types and sizes for control and dummy sections of the ROMs.

3. Expressions for definitions and references (primary, secondary, and forward references).

The second pass forms the actual core image and its relocation dictionary. (Reference: CP-V/BP Reference Manual, 90 17 64.)

DELTA

Delta is designed to aid in the debugging of programs at the assembly-language or machine-language levels. It operates on object programs and tables of internal and global symbols used by the programs but does not require that the tables be at hand. With or without the symbol tables, Delta recognizes computer instruction mnemonic codes and can assemble machine-language programs on an instruction-by-instruction basis. The main purpose of Delta, however, is to facilitate the activities of debugging by

1. Examining, inserting, and modifying such program elements as instructions, numeric values, and coded information (i.e., data in all its representations and formats).

2. Controlling execution, including the insertion of breakpoints into a program and requests for breaks on changes in elements of data.

3. Tracing execution by displaying information at designated points in a program.

4. Searching programs and data for specific elements and subelements.

Although Delta is specifically tailored to machine language programs, it may be used to debug any program. Delta is designed and interfaced to the system in such a way that it may be called in to aid debugging at any time, even after a program has been loaded and execution has begun. (Reference: CP-V/TS Reference Manual, 90 09 07.)

FORTRAN DEBUG PACKAGE

The FORTRAN Debug Package (FDP) is made up of special library routines that are called by Xerox Extended FORTRAN IV object programs compiled in the debug mode. These routines interact with the program to detect, diagnose, and in many cases, repair program errors.

The debugger can be used in batch and on-line modes. An extensive set of debugging commands are available in both cases. In batch operation, the debugging commands are included in the source input and are used by the debugger during execution of the program. In on-line operations, the debugging commands are entered through the terminal keyboard when requested by the debugger. Such requests are made when execution starts, stops, or restarts. The debugger normally has control of such stops.

In addition to the debugging commands, the debugger has a few automatic debugging features. One of these features is the automatic comparison of standard calling and receiving sequence arguments for type compatibility. When applicable, the number of arguments in the standard calling sequence is checked for equality with the receiving sequence. These calling and receiving arguments are also tested for protection conflicts. Another automatic feature is the testing of subprogram dummy storage instructions to determine if they violate the protection of the calling argument. (Reference: FDP/Reference Manual, 90 16 77.)

COBOL ON-LINE DEBUGGER

The COBOL On-line Debugger is designed to be used with Xerox ANS COBOL. The debugger is a special COBOL run-time library routine that is called by programs compiled in the TEST mode. This routine allows the programmer to monitor and control both the execution of his program and the contents of data-items during on-line execution. The debugger also allows the COBOL source program to be examined and modified.

The debugger can only be used during on-line execution; however, programs that have been compiled for use with the debugger may be run in the batch mode. This is not recommended, though, because of the increased program size when the TEST mode is specified. (Reference: ANS COBOL/On-line Debugger Reference Manual, 90 30 60.)

SERVICE PROCESSORS

The processors in this group perform general service functions required for running and using the CP-V system.

EDIT

The Edit processor is a line-at-a-time context editor designated for on-line creation, modification, and handling of programs and other bodies of information. All Edit data is stored on disk storage in a keyed file structure of sequence numbered, variable length records. This structure permits Edit to directly access each line or record of data.

Edit functions are controlled through single line commands supplied by the user. The command language provides for insertion, deletion, reordering, and replacement of lines or groups of lines of text. It also provides for selective printing, renumbering records, and context editing operations of matching, moving, and substituting line-by-line within a specified range of text lines. File maintenance commands are also provided to allow the user to build, copy, merge, and delete whole files. (Reference: CP-V/TS Reference Manual, 90 09 07.)

PERIPHERAL CONVERSION LANGUAGE

The Peripheral Conversion Language (PCL) is a utility subsystem designed for operation in the batch or on-line environment. It provides for information movement among card devices, line printers, on-line terminals, magnetic tape devices, disk packs, and RAD storage.

PCL is controlled by single-line commands supplied through on-line terminal input or through command card input in the job stream. The command language provides for single or multiple file transfers with options for selecting, sequencing, formatting, and converting data records. Additional file maintenance and utility commands are provided. (References: CP-V/TS Reference Manual, 90 09 07 and CP-V/BP Reference Manual, 90 17 64.)

SYSGEN

SYSGEN is made up of several processors. These processors are used to generate a variety of CP-V systems that are tailored to the specific requirements of an installation. The SYSGEN processors are PASS2, LOCCT, PASS3, and DEF. PCL is used to select from various sources the relevant modules for system generation. PASS2 compiles the required dynamic tables for the resident monitor. LOCCT and PASS3 file away and execute load card images to produce load modules for the monitor and its processors. DEF writes a monitor system tape that may be booted and used. (Reference: Chapter 14.)

DEFCOM

DEFCOM makes the DEFs and their associated values in one load module available to another load module. It accomplishes this by using a load module as input and by producing another load module that contains only the DEFs and DEF values from the input module. The resultant load module of DEFs can then be combined with other load modules. DEFCOM is used extensively in constructing the monitor and the shared run-time libraries. (Reference: CP-V/BP Reference Manual, 90 16 64.)

SYMCON

The Symbol Control Processor (SYMCON) provides a means of controlling external symbols in a load module and of building a global symbol table. Its primary function is to give the programmer a means of preventing double definitions of external symbols. It may also be used to reduce the number of external symbols. For example, if certain load modules cannot be combined because their control tables are too large, the tables may be reduced in size by deleting all but essential external symbols. (Reference: CP-V/BP Reference Manual, 90 17 64.)

ANALYZE

ANLZ provides the system programmer with a means of examining and anlyzing the contents of dumps taken during system recovery. It is called automatically by the Automatic Recovery Procedure and is executed as a ghost job. It may also be called by the operator to analyze tape dumps when recovery is not possible, or by an on-line user to examine dumps of the currently running monitor. (Reference: Chapter 9.)

BATCH

The Batch processor is used to submit a file or a series of files to the batch queue for execution. Through Batch processor commands, the following capabilities are available:

1. A file may be inserted into a file being submitted for execution, thus bringing together more than one file to create a single job.

2. Selected strings and fields existing in files being submitted for execution may be replaced by new strings and fields.

3. The results of string and field replacements can be examined before the job is submitted to the batch stream.

4. Files to be submitted for execution may reside on tape or private disk pack.

5. Jobs may be submitted to run in an account other than the account from which the job is submitted.

The Batch processor may be called in either the on-line or the batch mode. (Reference: CP-V/TS Reference Manual, 90 09 07.)

APPLICATION PROCESSORS

The application processors are intended for use for specific types of applications.

SORT/MERGE

The Xerox Sort/Merge processor provides the user with a fast, highly efficient method of sequencing a nonordered file. Sort may be called as a subroutine from within a user's program or as a batch processing job by control cards. It is designed to operate efficiently in a minimum hardware environment. Sorting can take place on from 1 to 16 keys and each individual key field may be sorted in ascending or descending sequence. The sorting technique used is that of replacement selection tournament and offers the user the flexibility of changing the blocking and logical record lengths in explicitly structured files to different values in the output file.

The principal highlights of Sort are as follows:

1. Sorting capability allows either magnetic tapes, disks, or both.

2. Linkages allow execution of user's own code.

3. Sorting on from 1 to 16 key fields in ascending or descending sequence is allowed. Keys may be alphanumeric, binary, packed decimal, or zoned decimal data.

4. Records may be fixed or variable length.

5. Fixed length records may be blocked or unblocked.

6. Disks may be used as file input or output devices, or as intermediate storage devices.

7. Sort employs the read backward capability of the tape device to eliminate rewind time.

8. User-specified character collating sequence may be used.

9. Buffered input/output is used.

(Reference: Sort-Merge/Reference Manual, 90 11 99.)

1400 SERIES SIMULATOR

The 1400 Series Simulator provides an economical and effective solution to the program conversion problem arising because of a change in hardware. This interpretive program is designed to execute 1400 series object programs automatically as if they were run on a 1401, 1460, or 1440. Thus, an existing level of computing capability can be maintained while new processing methods that take advantage of the new, more powerful Sigma equipment are designed and implemented.

The 1400 Series Simulator simulates object code produced by SPS, FORTRAN, Autocoder, RPG, and utility routines. Almost all 1400 operations may be simulated except for I/O operations in which hardware differences make total simulation impossible. Full 1400 operator capabilities are provided. (Reference: 1400 Series Simulator/Reference Manual, 90 15 02.)

EDMS (PROGRAM PRODUCT)[†]

EDMS is a generalized data management system that enables the user to create an integrated data base. It is designed to be used with COBOL, FORTRAN, and Meta-Symbol processors. It simplifies programming by performing most of the I/O logic and data base management for the application programmer.

The principal features of EDMS are as follows:

- The user can describe data in various data structures. Using sets, any element can be related to any other element. The data structures include lists and hierarchies (trees). The two relationships can be combined to form extensive networks of data.

- Access techniques include random, direct, indexed, and indirect (relative to another record).

- An EDMS data base may consist of up to 64 monitor files.

- Multiple secondary indexes can be defined by the user to allow records to be retrieved via any combination of secondary record keys.

[†]See "program product" in glossary.

- Users may construct any number of logical files or data bases within an EDMS file.
- Data is described separately from the user program to facilitate management of the data base.
- Comprehensive security exists at all levels of a file.
- Journalization provides an audit trail for backup and recovery.
- A dynamic space inventory is maintained to facilitate rapid record storage and to optimize the use of available storage space.
- Detailed data description is provided for inclusion into the user's application program to reduce programming effort.
- File I/O logic is performed for the user program including
  1. Logical or physical record deletion.
  2. Record retrieval on random or search basis.
  3. Record insertion or modification.

(Reference: EDMS/Reference Manual, 90 30 12.)

GPDS (PROGRAM PRODUCT)[†]

The General Purpose Discrete Simulator provides engineers and administrators, whose programming experience is minimal, with a system for experimenting with and evaluating system methods, processes, and designs. Providing a means for developing a broad range of simulation models, it allows organizing, modeling, and analyzing the structure of a system, observing the flow of traffic, etc. Potential applications include

- Advanced management planning.
- Analysis of inventory or financial systems.
- Studies of message switching and communications networks.
- Risk and capital investment studies.
- Evaluation and data processing systems.
- Job shop and queuing studies.

Although GPDS is compatible with other simulator systems, it has a number of salient features not usually found in competitive versions. (Reference: GPDS/Reference Manual, 90 17 58.)

CIRC (PROGRAM PRODUCT)[†]

CIRC is a set of three computer programs for electronic circuit analysis of Sigma 5-9 computers: CIRC-DC for dc circuit analysis, CIRC-AC for ac circuit analysis, and CIRC-TR for transient circuit analysis. The programs are designed for use by a circuit engineer at the installation, and require little or no knowledge of programming for execution.

CIRC can be executed with three modes of operation possible: conversational (on-line) mode, terminal batch entry mode, and batch processing mode. The system manager will determine which of these modes are available to the engineer, based on type of computer installation and other installation decisions.

- The on-line mode offers several advantages since it provides true conversational interaction between the user and computer. Following CIRC start-up procedures, CIRC requests a control message from the user. After the control message is input (e.g., iterate a cycle of calcuations with changed parameters) the computer responds (via CIRC) with detailed requests for application data. These requests are sufficiently detailed to virtually eliminate misunderstandings by the engineer. This mode is highly useful in a highly interactive environment that produces a low volume of output and requires limited CPU time.

- The terminal batch entry mode allows efficient handling of high volume output and large CPU time requirements while preserving the advantages of the terminal as an input device. Two files are required, one containing all CIRC input including a circuit description and control messages and the other directing the execution of CIRC. The job is entered from the terminal into the batch queue and treated like a batch job.

- The batch mode should generally be used for jobs involving large volumes of computations and outputs. It enables the user to concentrate on data preparation with virtually no involvement in programming considerations. The system manager can provide a set of start-up cards that never change, and these will constitute the entire interface between user and executive software. However, the batch mode offers less flexibility in experimenting with a circuit and slower turnaround time in obtaining answers.

(References: CIRC-AC/Reference Manual and User's Guide, 90 16 98, CIRC-DC/Reference Manual and User's Guide, 90 16 97, and CIRC-TR/Reference Manual and User's Guide, 90 17 86.)

[†]See "program product" in glossary.

USER PROCESSORS

Users may write their own processors and add them to CP-V or replace CP-V processors. The rules governing the creation and modification of processors are described in Chapter 7.

MONITOR

The monitor responds to the moment-by-moment requirements of controlling machine operation, switching between programs requiring service, and providing services at the explicit request of the user's program. The monitor programs that perform these functions are listed below.

1. Basic Control.
2. Scheduling and Swapping.
3. Memory Management.
4. File Management.
5. Multibatch Job Scheduling.
6. Resource Management.
7. Job Step Control.
8. Terminal I/O Handling.
9. Symbionts.
10. Cooperatives.
11. System Integrity.
12. Initialization and Start-up.
13. Operator Communications.
14. Batch Debugging.
15. Load-and-Link.
16. System Debugging.

The basic control system is an I/O interrupt service and handling routine. It includes trap and interrupt handlers, routines that place requests for I/O in a queue, and basic device I/O handling routines.

The scheduling and swapping module makes the decision to swap, selects the users to swap in and out, sets up the I/O command chains for swap transfers, and selects the next user for execution. It also ensures that any associated, but not currently resident, shared processors are brought in with each user. Special algorithms control I/O scheduling and the balance of machine use between on-line and batch.

The memory management module controls the use of core and disk storage. Specifically, it controls the allocation of physical core memory, maintains the map and access images for each user, services the "get" and "free" service calls for memory pages, and manages the swapping disk space.

File management routines control the content and access to physical files of information. These routines perform such functions as indexing, blocking and deblocking, managing of pools of granules on RADs and disk packs, labeling, label checking and positioning of magnetic tape, formatting for printer and card equipment, and controlling access to and simultaneous use of a hierarchy of files.

The multibatch job scheduling routines select jobs to be run from the waiting input queue depending on priority and resource and partition availability.

Resource management facilities keep track of the number of resources of each kind (i.e., tape drives, disk spindles, core) that are in use. For a batch job, the multi-batch scheduler compares the resources required with the available resources and does not start the job until sufficient resources are available. Once the job is started, the resources that are required by the job are reserved for the exclusive use of the job, thereby guaranteeing that they will be available for the duration of the job.

Job step control routines are entered between major segments of a job or an on-line session. They perform the monitor functions required between job steps such as

1. Processing error exit and abort CALs.
2. Handling monitor aborts.
3. Processing interpretive exits to associated shared processors or to load program modules.
4. Merging DCB assignments for execution.
5. Fetching program load modules into core.

Terminal I/O handling routines perform read-write buffering and external interrupt handling for I/O directed to user terminals. These routines also translate character codes, insert page headers and VFC control characters, simulate tabs, and perform other formatting tasks.

Symbiont routines transfer data from the card reader to logical device streams on disk storage and from logical device streams on disk storage to the card punch or line printer.

Cooperative routines intercept read, print, or punch commands in user programs and transfer data from or to logical device streams residing on disk storage. The input cooperative simulates card reading from a logical device stream. The output cooperative builds a logical device stream using intercepted program output directed by the user program to a line printer or card punch.

System integrity facilities provide error detection and recovery capabilities. This includes security to user files and automatic high-speed restart in case of system failure. Sufficient information is recorded to isolate errors and failures caused by hardware or software.

Initialization and start-up routines are stored on tape and are booted into core storage. After they are in core, they load the monitor root into core and turn control over to the root. The monitor root then completes the initialization of the monitor by starting and running the program called GHOST1 which completes the patching of the system and the initialization of the swapping disk and hardware.

Operator communication routines provide for communication between the monitor and the operator. They transmit messages to the operator and process key-ins received from the operator.

Batch debugging routines provide batch programs with debugging capability through the use of procedure calls. Any batch program may take a snapshot dump of a specified segment of memory, either on an unconditional or a conditional basis.

System debugging routines provide debugging services to system programmers. Three debugging routines are available. They are

1. Executive Delta: This is a stand-alone processor and is essentially the same as on-line Delta. Executive Delta is optionally loaded at boot time along with the root of the monitor and monitor system tables.

2. Analyze: This program is intended for debugging CP-V crash dumps. To accomplish this, it performs three major functions.

   a. It runs a series of monitor integrity checks to determine what caused the crash.

   b. It summarizes the complete software environment at the time of the crash in a series of tables.

   c. It permits on-line interactions similar to Delta.

3. Recover: This program provides the "bail-out" exit from the monitor. The error code that is transmitted to RECOVER defines the problem and the module that discovered the problem.

Load-and-link routines give batch programs three types of loading and linking capability. Through the use of procedure calls, a batch program may

1. Load an overlay segment into core storage.

2. Store the calling program on disk storage, load the called program into core storage, and transfer control to the called program.

3. Load a program into core storage, transfer control to the called program, and release the core area used by the calling program.

CP-V has two libraries. One is a public library and the other is a system library. In the standard release of CP-V, the public library contains two sets of programs. One set (P1) contains a useful set of Extended FORTRAN IV run-time library routines, the other set (P0) contains P1 and the FORTRAN Debug Package. These two libraries are so constructed that a single copy is shared among all concurrent users. The system library contains a collection of routines that are less frequently used than the public library routines. They are in library load module form and are loaded only with programs that reference them.

SCHEDULING AND MEMORY MANAGEMENT

Scheduling and memory management routines control the overall operation of the system. Inputs to these routines, together with the current status of users as recorded by the scheduler, are used to change the position of each user in the scheduling status queues. It is from these queues that selections are made for both swapping and execution. Swaps are set up by the selection of a high priority user that is to be brought into core and by pairing this user with one or more low priority users that are to be transferred to disk storage. Similarly, the highest priority user in core is selected for execution.

SCHEDULER INPUTS

System activities are reported by direct entry to the scheduler, which makes changes to user status queues through a logical signaling table. The scheduler records inputs by changing the user status queues and other information associated with the user. In general, a table-driven technique is used. The received signal is on one coordinate and the current state of the user is on the other. The table entry thus defined names the routine to be executed in response to the given signal-state combination. Since the number of signals and states is large, the table technique aids in debugging by forcing complete specification of all the possibilities. Inputs to the scheduler are listed in Table 1. The scheduler also receives control at execution of each CAL issued by a user program that is requesting monitor service. These entries (Table 2), the special entries from the executive language processors, and entries from internally reported events drive the scheduling of the system.

Table 1. Event Inputs Received by Scheduler

Event	Meaning
E:ABRT	Operator aborted user.
E:AP	Associate shared processor with user.
E:ART	Activate real-time user. Interrupt has occurred.
E:CBA	COC buffer available.
E:CBK	Break signal received.
E:CBL	Number of output characters > SL:TB.
E:CEC	TEL request (ESC ESC, ESC Y, or Y^c).
E:CFB	Cannot find COC buffer.
E:CIC	Terminal input message complete.
E:CRD	Read terminal command received.
E:CUB	Number of output characters = SL:UB.
E:DPA	RAD page available.
E:ERR	Operator errored user.
E:IC	I/O complete.
E:IIP	I/O started and now in progress.
E:IP	Request permission to start I/O.
E:KO	User removed from core.
E:NC	Cannot get requested core pages.
E:ND	Cannot get requested disk page.
E:NOCR	User allowed to open or close file.
E:NSYMD	No symbiont disk space.
E:NSYMF	No symbiont file entry.
E:NQR	Enqueue release – resource available.
E:NQW	Enqueue – wait for resource.
E:OCR	Request permission to open or close file.
E:OFF	User has hung up telephone.
E:QA	User queued for access (e.g., for access to tape or disk pack).
E:QE	Quantum end.

Table 1. Event Inputs Received by Scheduler (cont.)

Event	Meaning
E:QFAC	No granules available for user.
E:QFI	Real-time user. Queue for interrupt.
E:QMF	Queue for I/O master function count too high.
E:SL	Sleep time for user.
E:SYMF	Symbiont file now available.
E:SYMD	Symbiont disk space now available.
E:UQA	User dequeued for access (e.g., for access to tape or disk pack).
E:UQFAC	ALLOCAT has refreshed granule stacks.
E:WU	Wake-up time for user.

Table 2. Service Request Input to Monitor

Source of Inputs	Service Request Entries
User program (through monitor service calls)	1. Terminal input/output request.
	2. Input/output service calls for RAD, disk pack, or magnetic tape.
	3. Wait request.
	4. Program exit (complete).
	5. Core request (for common, dynamic, or specific pages).
	6. Real-time services.
	7. Program overlay (load and link, load and transfer).
	8. Debug requests.
	9. Miscellaneous service requests.
Executive language processor	1. Name of system programs to be loaded and entered (implies deletion of any current program).
	2. Continuation signal.
	3. Special continuation address.
	4. Link load-and-go-exit.

SCHEDULER OUTPUT

The scheduling routine performs two major functions during the time it is in control of the computer. The first function consists of setting up swaps between main core memory and secondary disk storage in such a way that high priority users are brought into core to replace low priority users that are transferred to disk storage. The actual swap is controlled by an I/O handler according to specifications prepared by the scheduler. These specifications are prepared according to the priority state queues described in the next section. Given a suitably large ratio of available core to average user size (greater than 4), the scheduler can keep swaps and computing close to 100 percent overlapped.

The second function the scheduler performs consists of selecting a user for execution according to the priority state queues and the rules for batch processing. The rule is simple: the highest priority user whose program and data are in core is selected.

USER STATUS QUEUES

Status queues form a single priority structure from which selections for swapping and execution are made. The status queues form an ordered list with one and only one entry for each user. The position in queue is an implied bid for the services of the computer. As events are signaled to the scheduler, individual users move up and down in the priority structure. When they are at the high end, they have a high priority for swapping into core and for execution. When they are at the low end, they are prime candidates for removal to secondary storage. This latter feature – that of having a defined priority for removal of users to disk storage – is an important and often overlooked aid to efficient swap management. It avoids extraneous swaps by making an intelligent choice about outgoing as well as incoming users.

In addition to these primary functions, user status queues have other functions such as

1. Synchronizing the presence in core of the user program and data with the availability of I/O devices.
2. Queuing user programs to be "awakened" at a preestablished time.
3. Queuing requests for entry and use of processors.
4. Managing core memory.
5. Queuing requests for buffers either in core or on disk.
6. Queuing requests for nonresident monitor services.

A list of the status queues is given in Table 3.

Table 3. Scheduler Status Queues

State	Meaning
SRT	Real-time execute (0 ≤ priority ≤ X'BF').
SC0	Background execute (X'C0' ≤ priority ≤ X'F5').
SC1	Background execute (priority = X'F6').
SC2	Background execute (priority = X'F7').
SC3	Background execute (priority = X'F8')
SC4	Background execute (priority = X'F9').
SC5	Background execute (priority = X'FA').
SC6	Background execute (priority = X'FB').
SC7	Background execute (priority = X'FC').
SC8	Background execute (priority = X'FD').
SC9	Background execute (priority = X'FE').
SC10	Background execute (priority = X'FF').
STOB	Terminal output blocked in core. (More characters than the system limit are ready for typing.)
STOBO	Terminal output blocked. Not in core.
SIOW	I/O wait. Users waiting for an I/O that is in progress to complete.
SIOMF	Users blocked because I/O master function count (number of I/O operations in progress) has reached the system limit.
SW	Users waiting for a specified "wake-up" time.
SQA	Users waiting for service by RBBAT, the symbiont ghost.
SQR	Users in core and blocked for dynamic resource such as swapper page, COC buffer, symbiont disk page, symbiont table space, enqueued resource, service by ALLOCAT (for file granules), or file open or close.
SQRO	Same as SQR but not in core.
STI	Typing input and in core.
STIO	Typing input and not in core.
SQFI	Real-time user waiting for interrupt.

SCHEDULER OPERATION

To select users for execution, the scheduler searches down a list of the status queues for the first user in core memory. The highest priority user is served first. Interrupting users are served before those with an active input message (both of these take precedence over users with unblocked terminal output), then come on-line compute-bound users and finally, compute-bound batch jobs. Note that users in lower states have no current requests for CPU resources. Note also that as each user is selected for execution, the status queue of the user is changed to CU. When the quantum is complete, the highest priority queue the user can enter is the compute queue. Users that enter any of the three highest priority states receive rapid response but only for the first quanta of service. Thereafter, they share service with others in the compute queue.

A similar selection procedure is used to set up users for swapping. First, the highest priority user in the execution queue who is not in core is selected and his size requirement (including the requirement for shared processors not in core) is determined. Second, users are selected from the swapout queue until enough space is freed by these users and their shared processors to provide for the user selected for swap-in. If a single user in a state below SC10 (Table 3) can be found to swap out, then a single rather than a multiple swap is chosen. No swaps occur until a user that is not in core enters a high priority queue.

Two lists resulting from this selection are presented to the swapper. One list contains the user (or users) to be swapped out and the other contains the user to be swapped in. This latter list also contains the shared processors that must accompany the user and the current free core page list. When the scheduler selects users for swapping, it picks a high priority user to load into core and the lowest priority user to remove from core. Priorities are arranged from high to low, in order of increasing expected time before the next activation. This ensures that the users that are least likely to be needed are swapped out first, while the users most likely to require execution are retained in core. The swap algorithm operates so that compute users remain in core and use all available compute time, while the interactive users are swapped through the third core slot whenever the following three conditions exist:

1. There is room in core for three user programs.

2. Two users are computing steadily.

3. Many other users are doing short interactive tasks.

Table 4 shows the queue used for selection of users to be brought in for execution and the queue used for selection of users to be moved to disk.

Note that the queues CU, IOW, QRO, TOBO, TIO do not appear in either list. Thus, the users in these states are not selected either for execution or for swapping.

Two examples of typical interactive use are illustrative of the scheduling operation.

Table 4. Swap-In and Swap-Out Queues

Swap-In (and Execution) Queue	Swap-Out Queue
SRT	SW
SC0	STI
SC1	STOB
SC2	SQFI
SC3	SQA
SC4	SC10
SC5	SQR
SC6	SC9
SC7	SC8
SC8	SC7
SC9	SC6
SC10	SC5
	SC4
	SC3
	SC2
	SC1
	SC0
	SRT

The first example traces scheduling operations for a simple, short interactive user request. At the time the request is typed, the user is in the STI queue. His program, which has probably been swapped to disk storage, remains there until the COC routines receive an activation character. Receipt of this character is reported to the scheduler and causes a change in state of the user to the appropriate executable state (SC0-SC10). The scheduler finds a high priority user not in core and initiates a swap to remove a low priority user (if necessary) and to bring in the one just activated. On completion of the swap, the scheduler is again called and now finds a high priority user ready to run. The user's state is changed to CU, the program is entered, and the input command is examined by the reading program. The cycle in this example is completed by preparation of a response line and a request to the monitor for more input which changes the user's state to TI again, making him a prime candidate for removal to disk.

The second example illustrates an output-bound terminal program. This program moves through the state cycle STOB-SC-SCU as output is generated by the program. The COC routines signal when the output limit has been reached, thus causing the program to be delayed while output is transferred to the terminal. In a typical operation, four to six seconds of typing is readied in buffers each time the user program is brought into core and executed. During this typing time, the program is not required in core and the CPU resources can be given to other programs.

I/O SCHEDULING

I/O scheduling is designed to provide good service to I/O-bound users while keeping the CPU busy with compute-bound users. The intent is to make the fullest possible utilization of both the CPU and the I/O devices. The manner in which this is accomplished is described below.

A user that has been waiting for an I/O to complete (SIOW) is changed to an executable state at a priority slightly higher than a similar compute-bound user when the I/O completes. At that time, the execution scheduler interrupts the execution of the compute-bound user so that the I/O-bound user can execute. The I/O-bound user requires comparatively little CPU time before initiating another I/O request and returning to the SIOW state. The compute-bound user then resumes execution.

It should be noted that the scheduler automatically adapts to jobs that alternate between bursts of computing and bursts of I/O.

SWAP HARDWARE ORGANIZATION

Users are removed from core to a dedicated area of disk storage (or to several disks in large configurations) when core is required for higher priority users.

Bit tables are used to keep track of the availability of each sector on the disks. In these tables, a zero is used to indicate the sector is in use (usually assigned to a user) and a one is used to indicate the sector is available. Users are assigned a sufficient number of page-size sectors to accommodate their current use. The assignment is done in such a way that command chaining of the I/O can order the sectors to be fetched for a single user with minimum latency. That is, each user's pages are spread evenly over the set of available sectors on the disk to which he is dedicated so that data will be transmitted in every disk sector passed over when the user is swapped.

The records of disk sectors associated with each user are kept in the user's job information table (JIT), which is kept on disk when the user is not in core. The disk location of the JIT and the user's disk address are kept in core by the scheduler. The disk layout is such that sufficient time is available after the user's JIT arrives from the disk for the system to set up the I/O commands for the remainder of the user.

The amount of disk storage assigned to swapping is a parameter of SYSGEN. The number of on-line users that the system can accommodate is limited by the size of disk space allocated for swapping and the total size of active on-line users.

PROCESSOR MANAGEMENT

CP-V processors are considered shared processors when they are written in such a way that they are pure procedure and are described as such when they are added to the system. (User-associated data areas are initialized at first entry.) A shared processor has the following special characteristics:

1. Its name is known to TEL and it may be called by name.

2. It has dedicated residency on swap storage established at system initialization or via DRSP.

3. A single copy is shared by all requesting users.

MEMORY LAYOUT

The system makes full use of Sigma mapping hardware, access protection, and write locks in allocating available physical core pages to uses. Physical core pages are allocated to users at their request. Use of the map obviates the need for program relocation or physical moves. Full protection is provided for one user from another. All programs and the monitor itself are divided into procedure and data. The procedure area is protected by write-locks or access codes, or both, against inadvertent stores.

The central features of the use of write-locks to protect master mode programs are as follows:

1. The monitor operates with a key of 01 and may store in

   a. Its own data area (LOCK = 01).

   b. Any batch, on-line or shared processor core (LOCK = 01).

   It may not store in its own procedure (LOCK = 11).

2. Keys of 10 and 11 are never used, nor are locks of 00 or 10.

3. Write-locks are initialized only once at system start-up and are not changed thereafter except when running under control of Executive Delta where they are used to enable data breakpoints.

The access code on virtual memory pages controls references made by slave mode programs (user programs and shared processors). This code is retained in the JIT of each user and is loaded when the user gains control. Write access to JIT and other job context areas is given to TEL, CCI, and LOGON.

The layout of virtual memory that applies to user programs and ordinarily shared processors is shown in Figure 2. Allocation of the available area depends on the type of user that is running and the attributes of the load module to be executed. Allocation Type II is used when a core library or debugger is associated or when the load module to be executed has been built by Link. In all other cases, allocation of the available area is as shown in Type I for batch users, ghost jobs, and on-line users executing in the extended memory mode.

Core addresses shown are those appropriate for a typical system but more (or less) core may be established for the resident monitor at SYSGEN time depending on installation needs. More (or less) area may also be desirable for the library area and for the job context area to accommodate more buffers. These bounds may also be adjusted at SYSGEN time. The boundary at which the one-pass loader (Link) places the user program is also adjustable.

Virtual pages not currently allocated to the user are mapped into a resident monitor page that is write-locked, (the access code is set to no access). Thus, slave mode programs are denied access through the access code, and attempts to store at these cirtual addresses by a master mode program are protected by write locks.

A typical layout of physical memory is shown in Figure 3. Although this is similar to the actual layout, it should not be assumed to be exact.

SYSTEM INTEGRITY

The monitor has a number of routines that have been included to guarantee system integrity. The objectives of

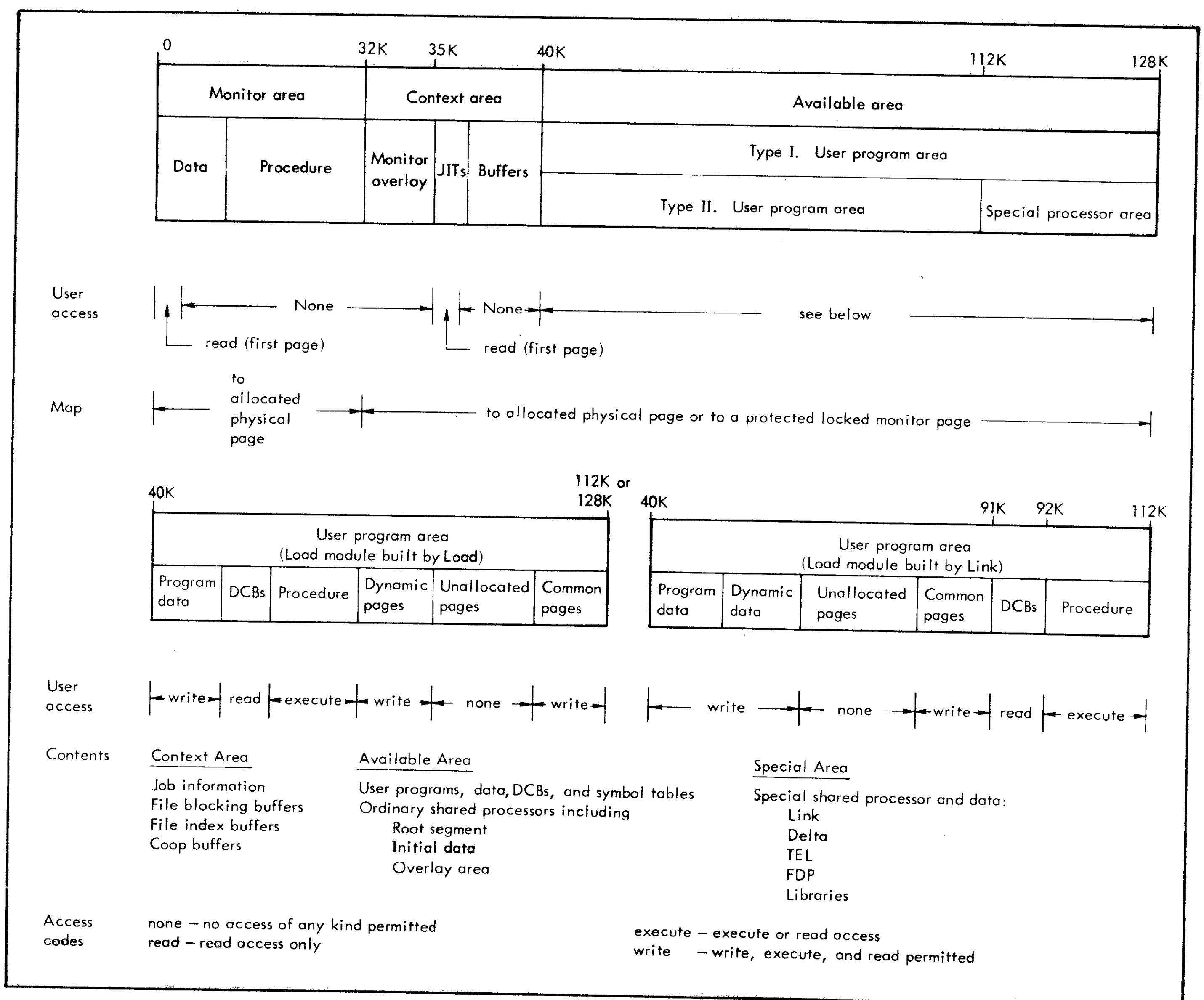


Figure 2. Typical User Program - Virtual Memory Layout (not to scale)

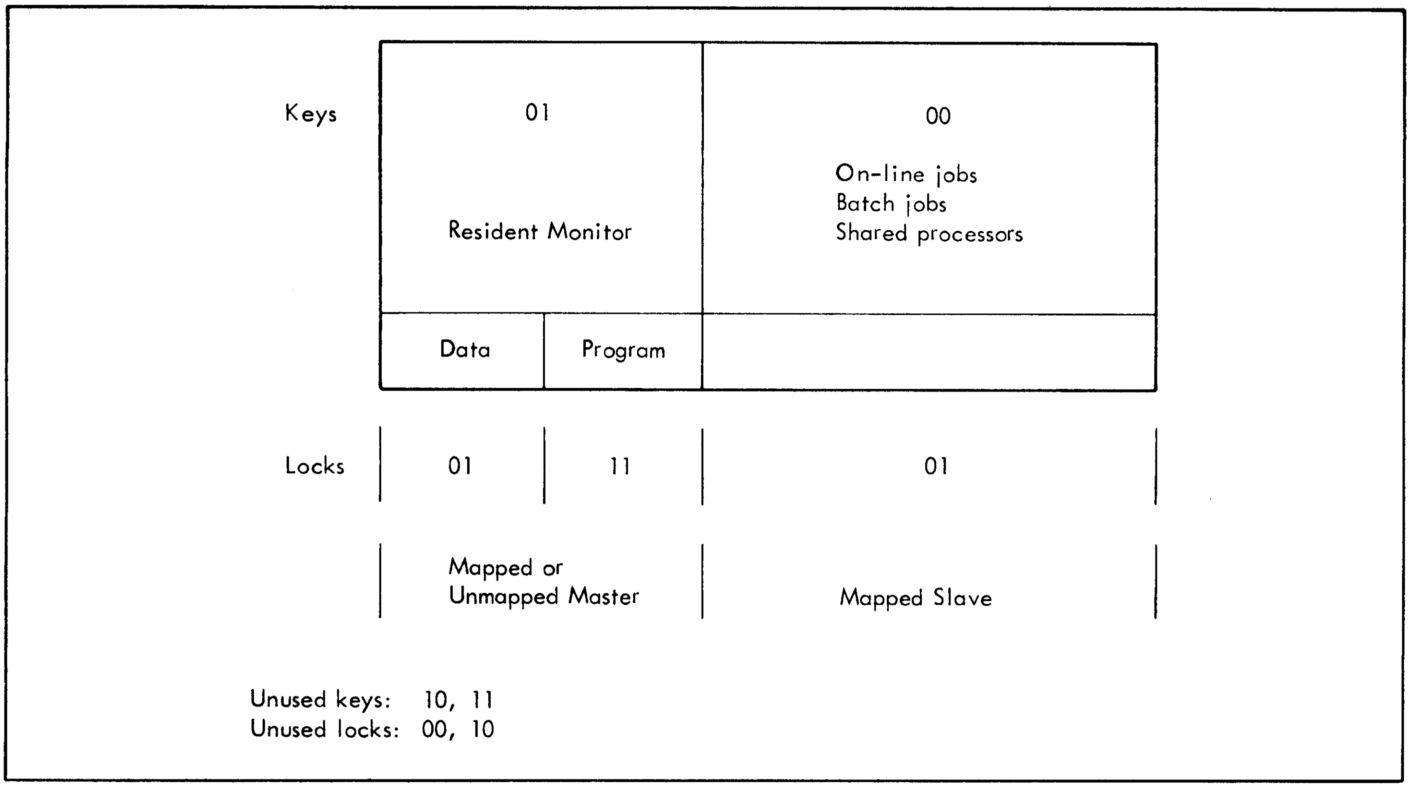


Figure 3. Typical Memory Layout (not to scale)

these routines are, in order of importance, (1) to provide the highest possible security for user files even in the event of total system failure, (2) to provide automatic high-speed recovery in the event of a machine or software failure, and (3) to record sufficient information to isolate errors and failures caused by either hardware or software.

The major features of the CP-V system integrity routines are as follows:

1. Detection of malfunctions by hardware examination and software checks wherever the checks have been shown to enhance hardware error detection. Recovery from these malfunctions is through retries, operator assistance, etc.

2. Logging of all malfunctions, including recovered errors and permanent failures.

3. Protection from hardware failures.

4. Use of on-line exercisers to provide for repair or adjustment of peripherals without taking the CPU down.

5. File backup and recovery facilities to minimize the probability of losing user files, and in case of file failure, to facilitate complete recovery of the file system with a minimum of loss.

6. Automatic recovery following a system failure with reasonable speed consistent with file security and the recording of information for later analysis.

7. Facilities to provide for analysis of system crashes. Information includes simple classification of failures as well as full information for both customer engineers and system programmers.

ERROR DETECTION AND RECOVERY

An effecitve operating system must be able to detect and, whenever possible, to correct errors. It must also be capable of restarting the system if necessary. CP-V uses a combination of hardware and software checks to efficiently meet these goals.

Hardware error protection features include memory protection against accidental overwriting of monitor and user programs, power fail-safe interrupts that ensure automatic restart in the event of power failure, memory parity checking, I/O read and write verification, and a watchdog timer to avoid instruction hangups. Detected errors are reported, logged, and if possible, recovered directly. Catastrophic failures cause an automatic system recovery if at all possible. Those failures which can be isolated to a single user cause only that user to be aborted.

Software consistency checks, some of which are performed optionally on the throw of a console sense switch, check the integrity of the software at many critical locations in the system. These checks detect problems before they are allowed to go beyond a recoverable point. When an inconsistency that is catastrophic to the system is detected, the current users are logged off and all open files are closed. The system is then automatically rebooted for the fastest possible restart.

ERROR AND FAILURE LOGGING

Malfunction messages are maintained in a special file by system integrity routines. Messages are placed in this file whenever malfunctions are detected by the various parts of the system. Hardware malfunctions that are recorded include such things as tape errors, card reader errors, memory parity errors, and illegal instructions. Software malfunctions that are recorded include the failure of software checks on RAD or disk addresses contained in index blocks and improper linkage of linked file blocks. In addition, a software recovery from a seek failure is recorded in this file (as a 757F code).

ERROR LOG ANALYSIS

The error log analysis program (ERR:FIL) is called into execution by the monitor whenever the number of malfunctions has exceeded a certain threshold. The analysis program has two major functions. One function consists of copying malfunction messages from a special file where they are initially recorded into a standard format file. The other function consists of producing time-trace summaries that are useful for predicting failure of devices. These summaries may be used by the Customer Engineer to aid in preventive maintenance of the system.

ON-LINE DIAGNOSTICS AND EXERCISERS

On-line diagnostics and exercisers may be called when there is a specific failure detected by the hardware or software, or when a failure is projected through analysis of the error log by the Customer Engineer. These programs may also be called by the Customer Engineer when needed for the test or adjustment of the card reader, card punch, line printer, magnetic tape, or other devices.

FILE MAINTENANCE

CP-V provides a variety of processors designed to maintain a reliable backup of the file data base. These processors are summarized in Chapter 8 and are described in detail in the CP-V/OPS Reference Manual, 90 16 75. The processors provide the ability to save and restore large volumes of files very quickly, to save and restore entire private and public disk devices at device speed, to handle user initiated backup of files, to restore the allocation tables for public disks after a system crash, to restore the allocation tables for a private disk pack after a crash which affected the pack, and to restore granule account information in the :USERS file.

AUTOMATIC RECOVERY AFTER SYSTEM FAILURE

The CP-V monitor performs consistency checks on the results of hardware operations, checks intermediate results of operating system software functions, performs checks and balances at appropriate interfaces between the operating system's modules, and monitors itself for unexpected trap conditions caused by the hardware or operating system software. A software check code is assigned to each type of failure that the monitor may detect.

Some of these software check failures result in a momentary delay in service to all but the current user for whom the operating system is performing a service. In such case, the current user's job step is aborted, core is dumped to a file for later analysis and display, and normal operating then continues. The remaining software check failures are handled by the system's recovery routine.

The recovery routine performs the following functions:

1. Displays cause of failure.
2. Takes a full core dump for later analysis.
3. Closes all open files with default options.
4. Packages or releases all partial symbiont files.
5. Packages error log.
6. Informs users of interruption.
7. Saves time, data, error log pointers, accounting information, symbiont file directory, public disk granule usage map, and executive communication.
8. Restarts system and restores items saved above.

When functions cannot be performed, they are noted on the operator's console. If the function is considered minor, recovery continues. If it is connected with file operations, the file identification is noted and recovery proceeds.

The recovery routine described above occurs automatically with a minimum delay (a few seconds) in system availability. Operator initiation of this recovery function is also allowed, providing for the event that the system fails by not responding to any operator key-in or user service request.

When the recovery routine executes, it is independent of all monitor services and functions and requires only that a small recovery driver be intact in memory. This driver reads the main recovery module into memory from the system swap device, overlaying the pure procedure portion of CP-V. Certain monitor system tables are also required intact for successful recovery. These tables are verified before proceeding. If the recovery process cannot be completed, the operator is instructed to initialize the system from the master system tape and restore files and backup tapes.

CRASH ANALYSIS

In the event of an operator initiated recovery, one of the functions of the recovery procedure is to dump the contents of core memory onto disk storage. This information is available for later analysis by system programmers and by a special program designed to print in readable form the contents of the monitor's control tables.

The crash analysis program is a privileged ghost program that is called automatically by the recovery routine as the first job following a hot start to analyze the last core image filed on disk. This program is written in such a way that additional tests may be included as they are found to be useful. Initially, it provides the following services and tests:

1. Prints PSD and register contents at point of error.
2. Prints direct cause of error.
3. Runs some of the same checks that are used to test the dynamic integrity of the system.
4. Prints the contents of the critical monitor tables.
5. Prints the contents of the current users JIT (which contains the active temp stack).
6. Prints the contents of the physical JIT.
7. Prints a hexadecimal dump of core memory.

The program may also be called from a console by the system programmer to examine the crash core image. Additional details about the program are included in Chapter 9.

3. RESOURCE AND LIMIT MANAGEMENT

CP-V provides an installation manager with extensive tools for defining and controlling the system's resources and services. The purpose of this chapter is to provide an overview of these tools with emphasis on their interrelationship.

RESOURCE MANAGEMENT

The term resource has a very specific meaning in the following discussion. A resource is any portion of the CP-V installation that is to be shared by the users in a manner such that each user requiring the resource is allocated the resource for its exclusive use. (An exception to this is private disk packs which under some circumstances may be shared even though they have been defined to be resources.) Peripheral devices and core are common types of resources. Symbiont devices and public storage devices can never be defined to be resources because they are non-allocatable devices; that is, they are never reserved for the exclusive use of one user.

There are special resource management routines within the monitor. The specific task of these routines is to keep track of the number of resources of each kind in use and the number of resources of each kind that are available for use. For a batch job, the requirement for resources is compared with the available resources and the job is not started unless sufficient resources are available. (The user specifies his resource requirements on the LIMIT control command.) Further, the resources are reserved for the exclusive use of the job so that it is guaranteed that they will be available even if a long time elapses between job startup and actual use of the resources.

The system manager must define what the resources are for the installation, establish system defaults and maximums for use of the resources, and set limits on the use of the resources for the individual users. He performs these tasks using the following processors:

- SYSGEN PASS2
- Control
- Super

In the PASS2 phase of System Generation, the system manager uses the :RES command to establish which portions of the installation are to be resources. For each resource, he establishes the amount of the resource that may be allocated to all concurrent batch jobs, to all concurrent on-line jobs, and to all concurrent ghost jobs. He also establishes the default amount that is to be allocated to each batch, on-line, and ghost job in cases where the amount is not otherwise specified and the maximum amount that may be allocated to individual batch, on-line, and ghost jobs.

The following types of resources are always defined at SYSGEN – if not explicitly, then by default:

CO – core

7T – 7-track tape drive

9T – 9-track tape drive

SP – disk pack spindle

A maximum of eleven more resources may also be defined.

The Control processor can be used to dynamically modify the default and maximum values associated with each resource. Resources must be defined at SYSGEN. New resources cannot be added to the system via the Control processor. However, a resource may be effectively removed from the system by appropriate modification of the values associated with the resource.

The Super processor is used to establish the maximum amount of each resource that is to be available to each user when the user runs in the batch or on-line modes. In special cases, an individual may be authorized a resource maximum which is higher than the system maximum to allow a special job to run when no other user can acquire that amount of resource. For example, the maximum for core could be set low during the day for pushing through a lot of small jobs, but an individul critical job could be run with a high core requirement.

LIMIT CONTROL

In order to coordinate the sharing of a CP-V installation among many users, it is necessary to impose limitations on the execution of user programs. These limitations fall into two categories:

1. Service limits which limit such things as:
   - Job execution time.
   - Pages of printer output.
   - Number of cards punched.
   - Amount of temporary public storage.
   - Amount of permanent public storage.
2. Resource limits which limit the number of resources of each type that are available for the job.

Limits are established, changed, and collected from four sources:

1. SYSGEN PASS2 processor – for establishment of system limit tables which define limits to be associated with each batch, on-line, and ghost job. These limits are established through use of the :RES, :BLIMIT, :OLIMIT, and :GLIMIT commands.

2. Control processor – for dynamic modification of the system limit tables.

3. Super processor – for establishment and dynamic modification of the limits for each individual user. The limits are recorded in the :USERS file, a file which contains one record for each user at the installation.

4. LIMIT control command – for establishment of limits on a particular instance of execution. (The LIMIT control command is only applicable to the batch mode.)

The sequence by which the ultimate service and resource limits are placed on an executing user program is depicted in Figure 4. When the job is started, limit values for the job are initially set from the :USERS file record. Values which are not given in that record are then set from the monitor limit tables. For batch jobs, limit values are reduced to the value specified by the LIMIT control command.

Finally, these composite values are compared to the maximum values in the SYSGEN/Control set monitor limit tables and the job is aborted if the limits are exceeded.

The process may be divided into two cases: first, when there is no user maximum specified in the :USERS file record for the limit in question, and second, when there is a user maximum specified. The algorithm applies both to service limits and to resource limits identically, except where noted.

Case 1: No User Maximum in :USERS file

The limit is set to the limit on the LIMIT control command if any. Otherwise, it is set to the system default. If the limit is less than or equal to the system maximum, the job is run. Otherwise, the job is aborted.

Case 2: User Maximum specified in :USERS file

If no LIMIT control command is included with the job, the limit is set to the user maximum for all service limits and all on-line resource limits. The limit is set to the user maximum or the system default (whichever is smaller) for batch resource limits and for job execution time.

If a LIMIT control command is included with the job, the limit is set to the limit on the command if it is less than or equal to the maximum specified in the :USERS file. Otherwise, the job is aborted.

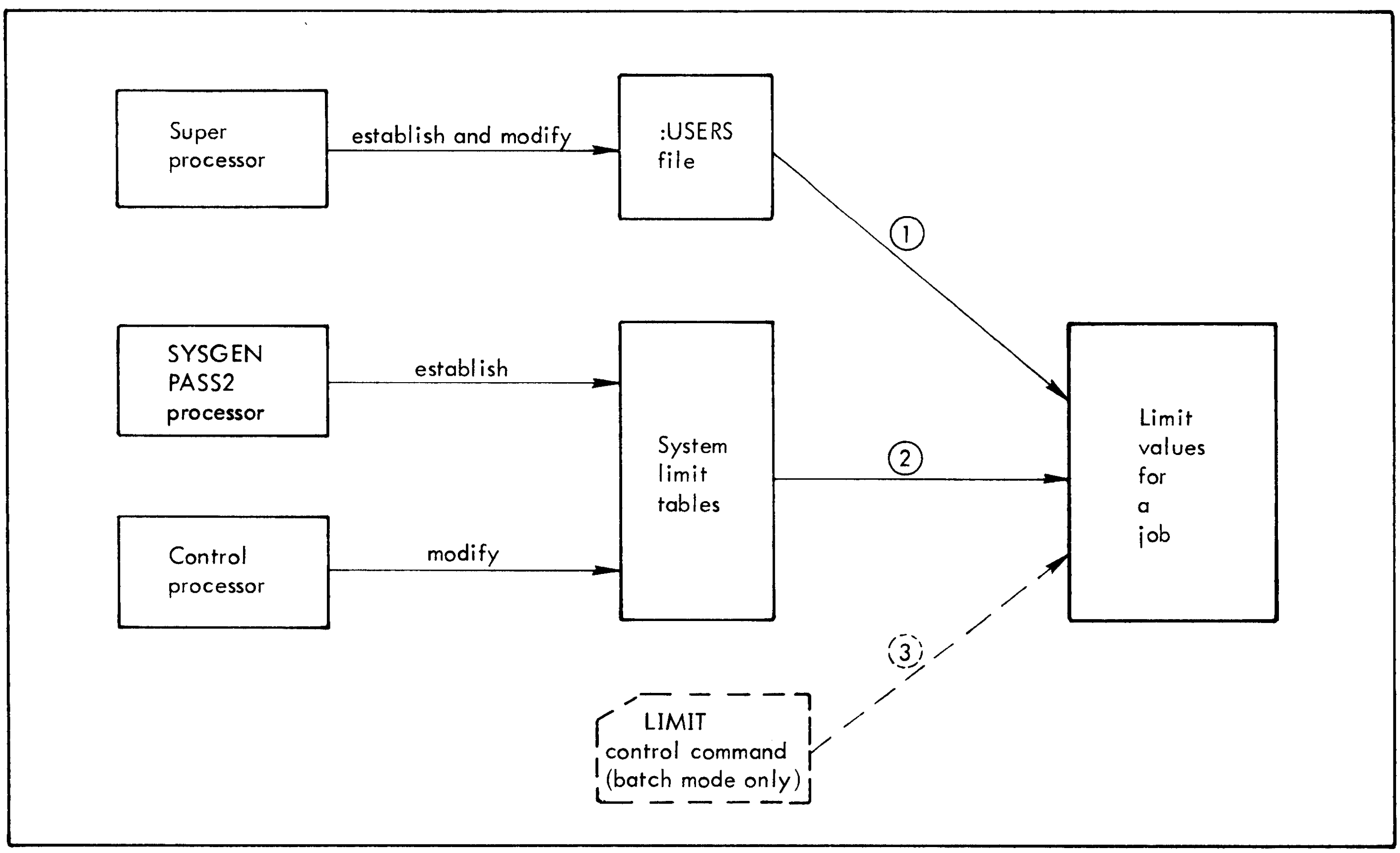


Figure 4. Establishing Limits for a Job

4. LOG-ON CONTROLS (SUPER)

INTRODUCTION

During log-on, four items are requested from the user: account, name, password, and extended accounting. (Password and extended accounting are optional.) These items are used to reference a log-on file (:USERS) that controls the entry of the job and, if the job is allowed, the type of usage and system privileges extended to the user. The log-on file is created by LOGON or CCI and is maintained by a specially authorized program, called Super, which may be run in the batch job stream or from an on-line terminal. Super must be run in the :SYS account (with any name) and the user must have at least CO privilege. Modifications to the log-on file are made using Super commands.

The log-on file exists in the :SYS account under the name :USERS. It is composed of a series of records, one for each user who is authorized to log on. Most of these records are created by the system manager using the processor Super. The one exception is the :SYS account with the user name LBE. The first time there is an attempt to log on under account :SYS and user name LBE, a record for this account and name is automatically generated and placed in the :USERS file. If this initial log-on does not include a password, the password record in the system record remains empty (contains zeros) and any on-line terminal may create and delete records in the file with Super after logging on under account :SYS and user name LBE. Thus, the initial log-on should also specify a password.

Records within the log-on file are keyed records with the key formed by the concatenation of account number and name of each valid user. Each record contains the identifying information, a password (which may be changed by the PASSWORD command) and other information that controls the system facilities granted to the user.

Super is also used to create and maintain the :RBLOG file which contains remote processing workstation authorizations. The records within this file contain information such as workstation name, type of remote terminal to be used at the workstation, maximum priority for jobs submitted from the workstation, and remote peripheral devices to be associated as part of the workstation.

SUPER COMMANDS

Super has nine commands and associated options. The commands are

CREATE
MODIFY
DEFAULT
LIST
REMOVE
FAST
WORKSTATION
X
END

Super prompts for commands with a single dash and for options with a double dash.

CREATE The CREATE command adds a new record to the log-on file. The format of the command is

$$\text{C[REATE]} \begin{Bmatrix} \text{account,name} \\ \text{account(name)} \end{Bmatrix}$$

where

account — is the account under which the record is created. Account may be from 1 to 8 characters in length and may be any of the following characters:

A-Z a-z 0-9 ← $ * % : # @ -

name — is the name of the user for whom the record is being created. Name may contain from 1 to 12 characters and has the same character set as account.

Only one account and name may be specified for each CREATE command.

Options for CREATE are entered into the terminal following the prompt for options. Each option name is followed by an equal sign and the value of the option that is to appear in the record. Options on the same line are separated by a semicolon. If no options are desired, a carriage return character must be entered following the initial prompt for options. This creates a record containing name, account, and default privilege and billing values. The end of a group of options is also terminated by entry of a carriage return character following a prompt for options. Options for the CREATE command are shown in Table 5.

When an option is not specified, the system default for that option is used at run-time. (The system default is not entered in the log-on record except in the case of privilege and billing.)

Four of the options are preceded by

$$\begin{Bmatrix} \text{B} \\ \text{O} \\ \text{G} \end{Bmatrix}$$

where B stands for batch mode, O stands for on-line mode, and G stands for ghost mode. (However, the user authorization record is currently not used for jobs running in the ghost mode.)

Example:

Assume that a log-on file record is to be created under account ENGNR and name DEVEL. This record is to have the password A321B6 and a charge class of 5 for the batch mode. The BASIC processor is to be automatically called when the user logs on.

```
-CREATE ENGNR,DEVEL (RET)
--PASSWORD=A321B6;CALL=BASIC (RET)
--B$BILLING=50 (RET)
-- (RET)
-
```

Table 5. CREATE Command Options

Option	Description
PA[SSWORD]= {identification NONE}	"identification" is the user password that is to appear in the record. Maximum length: 8 characters. The following characters may not be used: ? △ = / . > < ; , # ( ). NONE clears the password field to zero.
RE[AD]= {ALL NONE}	specifies the default for READ accounts for all files created by this user. Initial default is ALL. ALL sets the field to 0. NONE sets the field to 1. (The field is one bit in length.)
CA[LL] = {name[.[account][.password]] NONE}	"name", "account", and "password" form the load module file identification of the system processor to be automatically connected to the user terminal when the user logs on. Maximum length: name 11 characters account 8 characters password 8 characters If "name" alone is specified (i.e., name of a system processor), Super supplies :SYS as the account. The form name. (Note the trailing period.) is a convenient shorthand way of specifying that the load module is to come from the user's log-on account. The form name..password specifies that the load module comes from the user's log-on account and has a password. NONE clears the call name, account, and password fields to zero.
MA[XEXPIRE] = {days,hours NEVER NONE}	"days,hours" specifies the maximum period of time the files are to be retained. Maximum: X'FFFF'. NEVER sets the field to X'FFFF'. NONE clears the field to zero and is the default.
DI[SK] = {pgranules NONE}	"pgranules" specifies number of granules of permanent disk pack file space allowed the user. Maximum: $2^{31}-1$. NONE clears the field to zero.
{B O G} $BI[LLING]=charge	"charge" is the user charge class for accounting and may range from 0 to 7. The value specified by charge is a pointer to one of the eight charge rate tables in the :RATE file (see Chapter 5). The defaults are 0 for batch, 1 for on-line, and 1 for ghost.
{B O G} $PR[IVILEGE]=level	"level" is privilege level granted the user. Privilege codes are Hex Code — Privileges Allowed E0 — Utilize real-time services. C0 — Bypass security and account checks and issue M:SYS CAL. B0 — Access and change the monitor. A0 — Read and write error file; request devices; invoke diagnostic. 80 — Examine (but not change) the monitor. 40 — Default privilege level for batch and on-line.

Table 5. CREATE Command Options (cont.)

Option	Description
{B, O, G}Mname={value, DELE[TE]}	"value" specifies, in decimal, the maximum value allowed for the resource or service indicated by "name". DELETE specifies that the entry in the log-on record for the resource or service indicated by "name" is to be deleted and that system defaults are to be used instead. The "name" (immediately following M) must be either the name of a resource defined by the :RES command at SYSGEN (e.g., CO (core), 9T (tape), 7T (tape), SP (spindle)) or the name of a service. The services are listed in Table 6. (The "value" for core must be expressed in number of K words.)
{B, O, G}Pname={Y[ES], N[O], DELE[TE]}	specifies whether the peripheral device or special feature specified by "name" is available to the user (YES) or is not available to the user (NO). DELETE specifies that the entry in the log-on record for the peripheral device or special feature specified by "name" is to be deleted and that the system default is to be used instead. The "name" (immediately following P) must be the device type of a peripheral that has been defined to be a symbiont device via the :SDEVICE command during SYSGEN or a special feature that was defined via the :FAUTH command during SYSGEN. The default is to allow access to all symbiont devices.
XA[CCT]=character string	"character string" specifies installation-specific account information. A maximum of 24 characters is allowed. A semicolon will automatically terminate the field; i.e., the semicolon and the characters following the semicolon will not be inserted into the field.
Note: If a value that is greater than the system parameter but within the range allowed by Super is input, this value will be accepted by Super but the user will be limited to the system parameter.	

Table 6. System Services

Service	Definition
TIME	The limit (in minutes) for job execution time.
LO	The number of pages of printer output from all shared processors involved in running a job.
PO	The number of punched card records produced in running a job.
DO	The number of pages of diagnostics produced in running a job.
UO	The number of pages of printed output from all the executing programs in a job.
TSTO[RE]	The number of granules of temporary RAD storage that may be used by a job.
PSTO[RE]	The number of granules of permanent RAD storage that may be used by a job.
TDIS[K]	The number of granules of temporary disk pack storage that may be used by a job.
PDIS[K]	The number of granules of permanent disk pack storage that may be used by a job.
FPOO[L]	The number of file blocking buffers to be allocated to a job.

MODIFY The MODIFY command changes the specified fields in an existing record of the log-on file. The format of the command is

M[ODIFY] {account,name | account(name)}

where

account is the account under which the record was created. Account may be from 1 to 8 characters in length.

name is the name of the user for whom the record was created. Name may be from 1 to 12 characters in length.

Options for the MODIFY command specify the fields of the record to be changed. The word NONE following an option name deletes the option value in the record. Options for the MODIFY command are the same as for the CREATE command (Table 5).

The user whose log-on record is modified may be running during the modification. Since Super changes the log-on record but does not change the JIT, the modification will not take effect until the user logs off and then logs on again.

Example:

Assume that a log-on file record under account ENGNR and name DEVEL is to be modified. The password is to be changed from A321B6 to 48ZMIBA and the on-line charge class is to be changed from 5 to 7.

```
-MODIFY ENGNR,DEVEL(RET)
--PASSWORD=48ZMIBA(RET)
--O$BILLING=7(RET)
-- (RET)
- (RET)
```

DEFAULT The DEFAULT command allows the system manager to change the default values for options of the CREATE command. (This command is intended to simplify the authorization of classes of users.) Changes are made only for the options specified. The format of the command is

D[EFAULT]

The selected options and their new default values are entered following the prompt for options. The rules for entering options are the same as for the CREATE command.

Once specified, the default values will remain in effect until overridden by a subsequent DEFAULT command or until Super is reloaded (which will bring in the assembled defaults). The default values set by DEFAULT apply to the CREATE command but not to the MODIFY command.

It is important to note that when the default value for an option is initially specified for one of the three job modes (B, O, or G), default values of zero are automatically generated for the other two modes. The default values for the remaining two modes may subsequently be explicitly specified, but in any case the system values are overridden. In the example below, OM9T is set to zero.

Example:

```
-DEFAULT(RET)
--BM9T=3;GM9T=1(RET)
--B$PR=80(RET)
-- (RET)
-
```

LIST The LIST command lists the contents of the log-on file. This command may be used to

1. List the entire log-on file.
2. List all records under a specified account.
3. List from one to three specific records under one account.

To list the entire file, the LIST command is entered into the terminal without parameters.

L[IST]

To list only the records in a specific account, the LIST command is entered into the terminal along with the account.

L[IST] account

where account is the account number of the records to be listed. Account is limited to 8 characters.

To list from one to three specific records under one account, the LIST command is entered into the terminal along with the account and names of users whose records are to be listed.

$$\mathrm{L[IST]}\left\{\begin{array}{l}\text{account,name}\,[\text{,name}\,[\text{,name}]]\\ \text{account(name}\,[\text{,name}\,[\text{,name}]])\end{array}\right\}$$

where

account is the account number to which the records belong. Account is limited to 8 characters.

name is the name of the user whose record is to be listed. Only three user names may be specified. Multiple names are separated by commas and must be under the same account. Each name is limited to 12 characters.

Any combination of options may be specified. For each record, the name, account, and value of each option is listed. For options that have three values associated (B, O, and G), all three values are printed in the order B, O, and G, separated by commas. Options are entered in the same way as they are for other commands except that only the option name is specified.

The options for the LIST command are

AL[L]

PA[SSWORD]

XA[CCOUNTING]

CA[LL]

$BI[LLING]

$PR[IVILEGE]

Mname

Pname

$

M

P

If $ is entered, then both BILLING and PRIVILEGE values are listed. If M is entered, then all entries for maximum values allowed for resources and services are listed. If P is entered, then all entries concerning peripheral permission are listed.

If all options are desired, the word ALL is entered in response to the option prompt.

```
- LIST(RET)
-- ALL(RET)
-- (RET)
```

(The contents of all records in the log-on file are listed here.)

-

On the other hand, if no options are desired, the word NONE or a carriage return character is entered, or the option NONE is entered. Then only the names and account numbers are listed.

```
-LIST
--
(The names and account numbers are listed here.)
-
```

Example:

Assume that a log-on record was created as follows:

```
-CREATE 1234ABCD,C36
--PASSWORD=SECRETX
--CALL=INITIAL.123ABC
--B$PRIV=40
--O$PRIV=80
--BM9T=2
--BMSP=0
--BPLP=Y
--OPLP=Y
--OMLO=50
--BMLO=100
--BMTIME=15
--
```

A listing of the contents of that log-on record would be

```
-LIST 1234ABCD,C36
--ALL
--
ID=1234ABCD      C36
PA= SECRETX
CA= INITIAL      123ABC
BI= 00  01  01
PR= 40  80  40
   M 9T   2,0,0
   M SP   0,0,0
   M LO   100,50,0
   M TIME 15,0,0
   P LP   Y Y N
```

REMOVE The REMOVE command deletes a record from the log-on file. The format of the command is

$$R[EMOVE]\begin{Bmatrix}\text{account, name}\\ \text{account(name)}\end{Bmatrix}$$

where

account is the account number of the record to be deleted. Account is limited to 8 characters.

name is the name of the user whose record is to be deleted. Name is limited to 12 characters.

Only one account and name may be specified in each REMOVE command.

Example:

Assume that the record for user ABLE, account 8634 is to be deleted.

```
-REMOVE 8634,ABLE
-
```

FAST The FAST command causes a large batch run to be executed in a much shorter time than in normal mode. The format of FAST is

F[AST]

The FAST command is valid only in batch mode and should be used only if there are no other users in the system. The :USERS and :RBLOG files will be kept open (and hence can not be accessed) until exit from Super.

WORKSTATION The WORKSTATION command is used to authorize a remote workstation, to specify or change options for a particular workstation, and to list options for a particular workstation or for all workstations. The format of the command is

W[ORKSTATION] id

where id is a 1-8 character workstation name. At least one character must be alphabetic. If the workstation name is not the name of an authorized workstation, then a new workstation is being authorized.

Options for the WORKSTATION command are entered into the terminal following prompts for options. Options may be specified on the same line separated by semicolons or may appear on separate lines. When no further options are desired, a carriage return alone is entered following a prompt for an option.

The LW option (which causes the options of a workstation or of all workstations to be listed) is a special case. When the LW option is specified, no other options may be specified. In fact, when the LW option is specified, Super outputs the requested listing without prompting for further WORKSTATION options.

General options for the WORKSTATION command are listed in Table 7. The column TYPE OF TERMINAL specifies which type of terminal (RBT, 2780, or IRBT) the option is applicable for. Table 8 lists the device options which are used to define attributes of peripheral devices at IRBTs. These options apply only to the device specified on the DEV option that precedes them. The TYPES column specifies whether the options are legal for input devices (I), output devices (O), or both (I, O).

The industry recognizes several "standard" IRBTs (e.g., COPE 1200 and IBM 360/20 with IRBT software). These standard IRBTs have identical attributes including identical types of peripheral devices. When defining a standard IRBT, the option TYPE=STND may be specified and no further options will

Table 7. General Options of the WORKSTATION Command

Option	Type of Terminal	Description
LW[=ALL]	RBT, IRBT	Requests that the workstation definition be listed for the workstation specified by id. If =ALL is specified, all workstation definitions will be listed regardless of the id specified for the command. If the LW option is specified, it must be the only option that is specified for the command.
TYPE=type	-	Specifies the type of terminal where type may be 7670 – Xerox 7670 RBT 2780 – IBM 2780 RBT IRBT – IRBT STND – Standard IRBT (No other options can be specified.) The default type is 7670.
{SY[STEM] / NS[YSTEM]}	RBT, IRBT	Specifies whether or not jobs in the :SYS account may be submitted from this workstation. The default is NSYSTEM.
RP=n	RBT, IRBT	Specifies the maximum priority for jobs submitted from this workstation or the maximum priority for files that are being passed directly to an output device (see the DC option in Table 8). The default value is 7.
GJOB=name	RBT, IRBT	Specifies the 1-7 character name of a ghost job in the :SYS account that is to be started when this terminal logs on.
{MRB / NMRB}	2780 RBT	Specifies whether the terminal sends and receives single records (NMRB) or 400-byte multiple record blocks (MRB). The default is NMRB.
{NEM / EM}	RBT	Specifies whether EM characters are to be punched (EM) or are not to be punched (NEM) into cards at the RBT. The default is EM. Punching EM characters increases transmission speed during input of the cards but may make the cards unusable at non-RBT card readers.
LPP=value	RBT	Specifies, in decimal, the lines per page for the RBT printer. The default is 39.
MLP=value	RBT	Specifies, in decimal, the maximum line length for the RBT printer. The default is 120 for the 2780 RBT and 128 for the 7670 RBT.
MCP=value	RBT	Specifies, in decimal, the maximum length of cards punched at the RBT. If the NEM option is specified, this value is ignored. The default is 80.
{MS[T] / SL[V]}	IRBT	Specifies whether the CP-V system will act as the central site (master) or as an IRBT (slave) when this workstation is connected. The default is MST.
DS[M]=mask	IRBT	Specifies, in hexadecimal, a device selector mask used to separate device type from device number. The value specified for mask can range from 0 to FF. The default is F. (See "Device Selection" below.)
X1	IRBT	Specifies that the IRBT is capable of receiving multiple control records and data records within the same transmission block. (The software of the particular IRBT determines whether this is possible.)

Table 7. General Options of the WORKSTATION Command (cont.)

Option	Type of Terminal	Description
N1	IRBT	Negates an X1 option and is only used when changing the attributes of a workstation. If X1 was never specified for the workstation, N1 is meaningless.
X2	IRBT	Specifies that the workstation is another CP-V system.
N2	IRBT	Negates an X2 option and is only used when changing the attributes of a workstation. If X2 was never specified for the workstation, N2 is meaningless.
RM[T] = nn	IRBT	Specifies a two-character remote number. This specification is only valid when CP-V is acting as a slave IRBT to another computer system. The number is assigned by the system manager of the other computer system.
RW[SN] = xxxx	IRBT	Specifies a one- to eight-character WSN to be used by the CP-V system to identify itself when logging onto the remote station being defined. This specification is only valid when CP-V is acting as a slave to another system.
DEV = devname	IRBT	Specifies the name used on the LDEV command and remote batch control commands to reference a particular device at the workstation. OC is a reserved device name and should be used as the devname if the device is to be used as an operator's console. The OC device cannot be accessed by users. The DEV option is followed by a list of options that define the particular device (see Table 8). Each device of the workstation must be defined in this manner. A minimum of 1 and a maximum of 16 devices may be defined.
DD = devname	IRBT	Specifies that the named device is to be deleted from the workstation definition. Options of a given device cannot be changed individually. The device must be deleted and completely redefined.

Table 8. Device Options of the WORKSTATION Command

Option	Type	Description
{IN OUT}	-	Specifies whether records for this device are to come IN to or OUT of the CP-V system. Devices capable of input and output must be defined as two separate devices. The default is OUT.
RC[B] value	I, O	Defines, in hexadecimal, the RCB used to communicate with the device being defined. The RCB is a one-byte field that specifies the type of device and, in some cases, the number of the device. The required value for the RCB is defined by the software of the IRBT. This option is required because it establishes the connection between the device name specified (DEV=devname) and a particular physical device.
IR[CB] value	-	Defines, in hexadecimal, the RCB for the operator's console when it is being used as an input device. (The RCB option defines the RCB for the operator's console when it is being used as an output device.) The IRCB option is only valid when an operator's console is being defined.

Table 8. Device Options of the WORKSTATION Command (cont.)

Option	Type	Description
SU[SBIT]=value	O	Specifies, in hexadecimal, which bit in the Function Control Sequence field is used as the suspend control bit for this device (see the Multileaving appendix in the CP-V/RP Reference Manual, 90 30 26). The position of the bit that is set to one indicates which bit is the suspend control bit. All other bits are set to zero.
SR[CB]=x	I, O	Specifies the subrecord control byte type for this device. The values may be P for printer type C for card type U for user supplied If SRCB=U is specified for an output device, the user writing to that device will be expected to supply a subrecord control byte as the first byte of each data record. If it is used for an input device, the subrecord control byte will be passed to the user with each data record. SRCB=C must be specified for input control devices. The default value is C.
LI[ST]=x	O	Specifies one of the following: Y this is a listing device. The device will only be used when the user specifically requests it with the LDEV command. N this is not a listing device. S this is the system listing device and will be used as the default listing device unless the user specifically requests another listing device. P This is the system punch device and will be used as the default punch device unless the user specifically requests another punch device. The default is N.
{CT[L] / NC[TL]}	I	Specifies whether or not this device is a control device (i.e., whether or not input from this device is to be scanned for jobs and remote control commands). The default is CTL.
DC=type	I	Specifies the device type of the local symbiont device to which files from this device will be sent directly. The default is NONE.
{SM[D] / NS[MD]}	O	Specifies that this is the system message device (i.e., the device to which messages to the operator will be sent). Only one system message device may be defined for a workstation. The default is NSMD.
{BI[NARY] / NB[INARY]}	I, O	Establishes whether or not binary input or output is legal for this device. The default is NBIN.
MA[XREC]=n	I, O	Specifies, in decimal, that the longest record legal for this device may have n bytes. The value n may range from 1 to 255. The default value is 80 unless LIST=Y or LIST=S is specified, in which case the default value is 132.

Table 8. Device Options of the WORKSTATION Command (cont.)

Option	Type	Description
MI[NREC] n	I, O	Specifies, in decimal, that the shortest record legal for this device may have n bytes. The value n may range from 1 to 255. The default value is 1. If LIST=Y or LIST=S is specified, then MINREC specifies the number of lines per page (i.e., the maximum number of lines allowed per page). In this case, the value n may range from 1 to 255 and the default value is 38.
PR[IV] p	I, O	Specifies, in hexadecimal, the privilege level required to use this device. The default value is 40.
KEEP	I, O	Specifies that an output file for this device is to be kept intact until the entire file has been output. This allows the complete file to be retransmitted after a line loss. KEEP is only meaningful for input devices if the DC option is also used. In this case, partial input files at a line loss are deleted rather than being output. Input files that are not direct passed are always deleted in this situation.

be required (in fact, no other options should be specified). All attributes (including those for peripheral devices) will be automatically established by Super. The devices for standard IRBTs and some of the option values for those devices are listed in Table 9.

The various options are to some degree order-dependent; that is, certain options exclude selection of other options. An incorrect or inappropriate selection of an option will result in a diagnostic response from Super and the option will be ignored. Figure 5 partially clarifies this point. The figure lists all of the options for the WORKSTATION command (with the exception of LW because it is a special case). The level of indentation indicates the order in which the options should appear; i.e., options that are indented can not be specified unless the option under which they are indented has been previously specified. Within a given level of indentation, the order of appearance of the options is not important. For example, TYPE IRBT must appear before DSM, but DSM need not appear before X1. The figure does not point out that some options are mutually exclusive (e.g., MST and SLV). However, all cases of mutually exclusive options are obvious from the option descriptions.

Device Selection. Users refer to local or remote symbiont devices by name in the DEV option of the LDEV command. If the device is at a remote workstation, then the name of the device was established with the DEV option of the WORKSTATION command. This external name is linked to the actual physical device with the RCB option of the WORKSTATION command. The format and content of the RCB may vary, dependent upon the software of the IRBT. Standard RCBs specify a device type and the number of the device. The format of a standard RCB is

0	1 2 3	4 5 6 7
	Device no.	Device type

where bit 0 is always set to one.

For example, if a system using standard RCBs has two line printers, the two RCBs would be

Line printer 1

0	1 2 3	4 5 6 7	
1	0 0 1	0 1 0 0	(X'94')

Line printer 2

0	1 2 3	4 5 6 7	
1	0 1 0	0 1 0 0	(X'A4')

The DSM specifies a device selector mask that determines which bits of the RCB will be used to choose an explicit device. For a standard RCB, a DSM of X'0F' means that only device type will be used in selecting a device and that the number of the device will be ignored.

Table 9. Standard IRBT Device Option Values

Device Type	RCB	IRCB	SRCB	SUSBIT
OC	91	92	C	40
CR	93	--	C	--
LP	94	--	P	800
CP	95	--	C	1

```
WORKSTATION id
     SYS
     NSYS
     RP
     GJOB
     TYPE=7670
          EM
          NEM
          LPP
          MLP
          MCP
     TYPE=2780
          MRB
          NMRB
          EM
          NEM
          LPP
          MLP
          MCP
     TYPE=STND
     TYPE=IRBT
          DSM
          X1
          N1
          X2
          N2
          DD
          MST
          SLV
               RMT
               RWSN
          DEV=OC
               IRCB
          DEV=any device (including OC)
               PRIV
               RCB
               SRCB
               MAX
               MIN
               DIR
               NDIR
               BIN
               NBIN
               KEEP
               IN
                    CTL
                    NCTL
                         DC
               OUT
                    SUSBIT
                    LIST
                    SMD
                    NSMD
```

Figure 5. WORKSTATION Command Options

For example,

Assume that the following DSM and line printers have been defined:

DSM=0F	(or DSM=F)
DEV=LP	(line printer 1)
RCB=94	
DEV=PR	(line printer 2)
RCB=A4	

If the user requests the device LP, he will get either LP or PR (line printer 1 or line printer 2), whichever device is available. The request for LP is translated to RCB=94, but the DSM specifies that only the low-order four bits (type) are to be used. Therefore, no distinction is made between device number one and device number two.

A DSM of FF means that the entire RCB is to be used in selecting the device. In the example above, a DSM of FF would mean that the user would get the device LP when he specified LP and PR when he specified PR.

A DSM of 00 means that the RCB is to be ignored completely in selecting a device, regardless of the device name specified by the user.

Note that the setting of the first bit in the DSM is not important because the first bit of the RCB is always set to one (even if the RCB is not a standard RCB) and the first bit of the DSM is effectively ignored. Therefore the following DSMs are equivalent:

00 and 80
0F and 8F
7F and FF

Note also that the bit settings of the DSM have a different meaning for RCBs that have a format other than the standard format. Generally speaking, however, the role of the DSM is the same for all RCB formats. It specifies which bits of the RCB will be used to select a device.

Examples:

1. Assume that workstation STA1 with three devices (an operator's console, a card reader, and a line printer) is to be defined:

```
-WORKSTATION STA1 RET
--TYPE=IRBT RET
--DEV=OC RET
--RCB=91 RET
--IRCB=92 RET
--SMD RET
--SUSBIT=40 RET
--DEV=CR RET
--IN RET
--RCB=93 RET
--DEV=LP RET
--RCB=94 RET
--SUSBIT=800 RET
--SRCB=P RET
--MAXREC=132 RET
--LIST=S RET
-- RET
```

2. Assume that a standard workstation with the workstation name STANDARD is to be defined and that the attributes of the workstation are then to be listed:

```
-W STANDARD
--TYPE=STND

-W STANDARD
--LW

ID=     4 STANDARD
TYP=IRBT
MODE=MSI
DSM=    0F
RP=     07
IRCB=   92
SMD=    0C
DEVICES      4
```

DEV	RCB	SRCB	SUS	I/O	LIST	CTL	BIN	KP	PRV	DC	MAX	MIN
OC	91	C	0040	OUT	N	N	N	N	40	00	80	1
CR	93	C	0000	IN	N	Y	N	N	40	00	80	1
LP	[illegible]	P	0800	OUT	S	N	N	N	40	00	132	38
CP	[illegible]	C	0001	OUT	P	N	N	N	40	00	80	1

X The X command deletes the definition of a workstation from the :RBLOG. The format of the command is

X id

where id specifies the workstation name of the workstation definition to be deleted.

END The END command causes an exit from Super to TEL. The format of END is

E[ND]

Example:

Assume that the PASSWORD, PRIVILEGE, and BILLING options for users ABLE and BAKER under account 8634 are to be listed and that the record for user ABLE is to be deleted. Upon completion, control is to be returned to TEL.

```
-LIST 8634,ABLE,BAKER
--PASSWORD
--FLAGS
--$
--
:  (listing)
-REMOVE 8634,ABLE
-END
!
```

BATCH OPERATIONS (SUPER)

Super can be run in a batch job stream to facilitate and speed up the modification of the :USERS file. The formats of the commands and command options are very similar to the on-line commands and options. If the user does not assign M:SI and M:LO to other devices, Super reads the commands and options from the card reader and writes message output on the line printer. Should the user assign M:SI to some other device (e.g., a magnetic tape), the input format must conform exactly to that prescribed below for card input.

Super is called by a !SUPER command. After it has been invoked, it reads the SI device for command input. Super commands are keypunched beginning in column 1. Command options are keypunched beginning in column 2. Figure 6 depicts a sample Super batch job deck.

Use of the FAST command will greatly speed up a large batch run.

In batch operation, default message output goes to the system line printer. When initially invoked, Super advances to a new page on the printer. It prints each command option as received, before it analyzes the input for correct syntax. When it detects a LIST command, it advances to a new page before writing the command on the printer. The LIST command print-out on the line printer has a format similar to the LIST print-out on a terminal but has just one line per record.

At the end of the batch Super run, two messages summarizing the number of errors are printed. They have the following format:

n COMMAND ERRORS

m USER FILE I/O ERRORS

The first indicates the number of syntax errors found in the input stream; the second indicates the number of abnormal conditions found in accessing the :USERS file. In general, a syntax error found in an option for a main command will result in the command being executed as though the particular option were not present.

SUPER ERROR MESSAGES

An error message will be returned to the terminal if a Super command is entered incorrectly. These error messages are listed in Table 10.

SUPER COMMAND SUMMARY

Table 11 summarizes Super commands. The left-hand column lists the command format, the right-hand column defines the command and options.

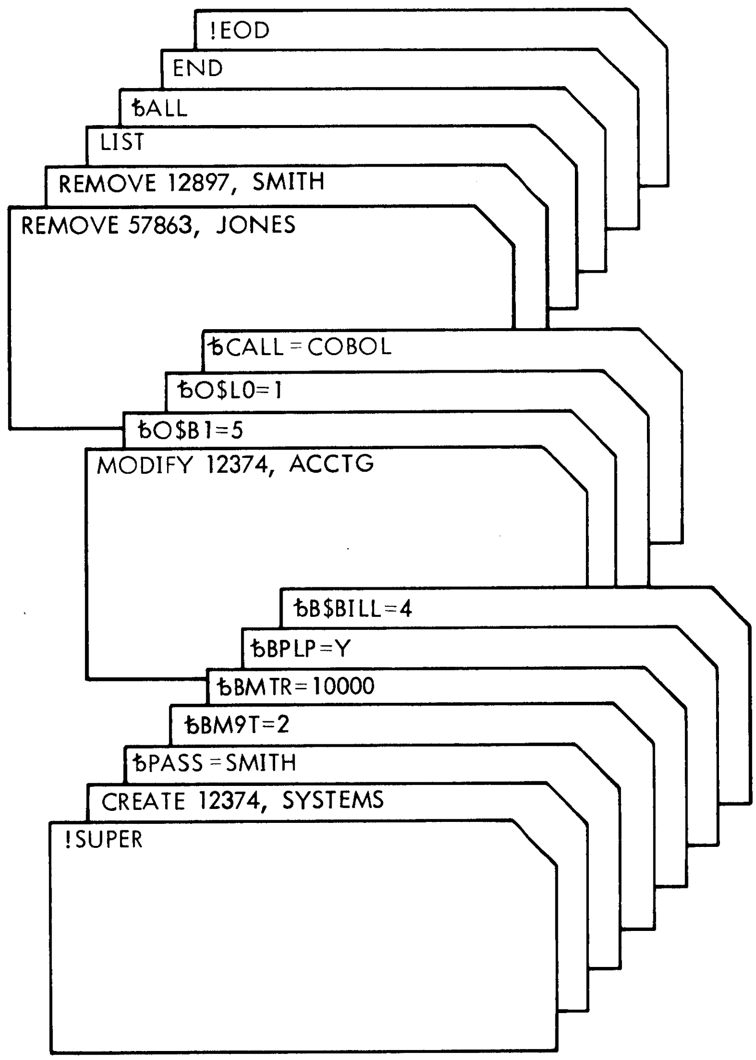


Figure 6. Sample Super Batch Job Deck

Table 10. Super Error Messages

Message	Description
ABNORMAL RETURN ON :USERS FILE -- value, value	An abnormal return other than "busy", "wrong key", or "file nonexistent" for I/O access of :USERS file occurred. The first value is a hex abnormal code; the second value is a hex abnormal subcode.
ACCOUNT, NAME ?	Super scanned to the end of the command buffer without detecting an account and/or name field for a CREATE, MODIFY, or REMOVE command.
ALREADY SPECIFIED FOR ANOTHER DEVICE	In the WORKSTATION command, the option LIST=P can only be specified for one device.
CALL ACCOUNT?	Super did not detect a delimiter (space, carriage return, period, nor semicolon) for a CALL lmn value.
DEVICE NAME ALREADY SPECIFIED IN RECORD	In the WORKSTATION command, a particular device name can only be specified once with the DEV option.
ERROR ON M:SI DEVICE, SUPER EXITING	An abnormal return other than EOF or EOD on the batch M:SI device occurred.
ERROR RETURN ON :USERS FILE -- value, value	An error return other than "wrong key" for I/O access of :USERS file occurred. The first value is a hex error code; the second value is a hex error subcode.
FAST COMMAND INVALID ON-LINE	The FAST command is only permitted as a batch command.
ILLEGAL CHARACTER IN HEX FIELD	A hexadecimal value was expected in an option of the WORKSTATION command. At least one of the characters in the value was not a hexadecimal digit.
ILLEGAL OPTION FOR SPECIFIED DEVICE	In the WORKSTATION command, an option that is not appropriate for the device being defined was specified or the device has not yet been specified but an option was specified for it.
ILLEGAL RCB VALUE	The low-order digit of the value specified on the RCB option of the WORKSTATION command cannot be a zero.
ILLEGAL SRCB VALUE	The value specified for the SRCB option of the WORKSTATION command must be U, P, or C.
ILLEGAL VALUE SPECIFIED	The value specified for the LIST option of the WORKSTATION command must be Y, S, N, or P.
IN/OUT CONFLICT	The mutually exclusive options IN and OUT were specified on the WORKSTATION command.
INVALID DEVICE NAME	The device name specified for the DEV option of the WORKSTATION command is not a name that was defined for the system at SYSGEN.
LOST AN OPTION; ADVISE LISTING USER	This message should never be issued; however, should it be, it indicates a Super program failure, or a loss of bits in the system. Super outputs the message and continues processing any remaining options in the buffer.
MASTER/SLAVE CONFLICT	The mutually exclusive options MASTER and SLAVE were specified on the WORKSTATION command.
MODIFY ?	A request was entered to CREATE a user (record) which already exists in the :USERS file. Super issues this message, then (prompts and) awaits input of options to modify the existing record. Null option input leaves the record as it was.

Table 10. Super Error Messages (cont.)

Message	Description
NO DEVICE SPECIFIED	In the WORKSTATION command, the workstation has been defined to be an IRBT but no DEV options are specified.
NO DEVICE SPECIFIED FOR DEVICE dd	In the WORKSTATION command, an RCB must be specified for each device defined.
NO SUCH DEVICE IN RECORD	In a WORKSTATION command, a device that does not exist is specified to be deleted.
NOT A SYMBIONT DEVICE	The device specified on the DC option of the WORKSTATION command must be a symbiont device.
NOT ALLOWED FOR TYP SPECIFIED	In the WORKSTATION command, an option was specified that is not allowed for the type of workstation (7670 or IRBT) being defined.
NOT MAIN COMMAND	Batch Super expected to read a command beginning in column one, but column one of the card was null.
RCB NOT YET SPECIFIED	The SMD option of the WORKSTATION command was specified before the RCB option was specified for a device, or a device was defined without an RCB being defined for it.
SMD ALREADY SPECIFIED	Only one system message device (SMD) is allowed per workstation.
SORRY YOU ARE NOT ALLOWED TO ACCESS SUPER	The user is trying to access Super and is not logged on under :SYS, LBE.
SPECIFIED VALUE TOO BIG	A value specified for an option on the WORKSTATION command is too big.
SPECIFIED VALUE TOO SMALL	A value specified for an option on the WORKSTATION command is too small.
SRCB INCONSISTENCY	An option of the WORKSTATION command is inconsistent with the value specified for the SRCB option.
STRING TOO LONG	The value specified for an option in the WORKSTATION command contains more characters than are allowed.
STRING TOO SHORT	The value specified for an option in the WORKSTATION command contains fewer characters than are required.
SYNTAX - UNEXPECTED EQUALS	An option on the WORKSTATION command contains an equal sign and an equal sign is not part of the particular option syntax.
THE :USERS FILE DOES NOT EXIST	This message should never be issued; however, should it be, it indicates Super, in accessing the :USERS file, received an abnormal return indicating :USERS does not exist. Super outputs the message and exits to executive level (TEL or CCI). Call Super again.
THIS OPTION ILLEGAL AT THIS TIME	An option on the WORKSTATION command was input before another option that must precede it was input.
TOO MANY OPTION NAMES, RE-DO LIST CMD	More than 16 individual options have been entered in either the M or P category on the LIST command.
UNRECOGNIZED NAME	The name specified on the M or P command is not valid.

Table 10. Super Error Messages (cont.)

Message	Description
UNRECOGNIZED TYPE NAME	The value specified for the TYPE option of the WORKSTATION command is not 7670, IRBT, or 2780.
VALUE TOO BIG	An option value was entered that is too big to fit in its designated field in the log-on record.
WARNING: NO SMD SPECIFIED	In the WORKSTATION command, the MASTER option was specified but no SMD option was specified.
WHO ?	A request was entered to MODIFY, REMOVE, or LIST the record(s) of a user(s) not in the :USERS file.
WORKSTATION NOT PRESENT	The user has attempted to delete a nonexistent workstation.
value?	Super does not recognize the given command operator or option operator, or does not recognize a too lengthy CALL field value. Input the information again.
=value?	The given option value is too long, exceeds system limit for the option, or contains an illegal character. Input the option again.

Table 11. Super Command Summary

Command	Description
C[REATE] {account,name / account(name)}	Adds a new record to the log-on file. Options: PA[SSWORD] = {identification (1-8 characters) / NONE} RE[AD] = {ALL / NONE} CA[LL] = {name[.[account][.password]] / NAME} name (1-11 characters) account (1-8 characters) password (1-8 characters) MA[XEXPIRE] = {days,hours / NEVER / NONE} DI[SK] {pgranules / NONE} {B / O / G} SBI[LLING] = charge ($0 \leq$ charge ≤ 7)

Table 11. Super Command Summary (cont.)

Command	Description
C[REATE] {account,name / account(name)} (cont.)	{B / O / G} $PR[IVILEGE] = level Hex Codes — Privileges Allowed E0 — Utilize real-time services. C0 — Bypass security and account checks and issue M:SYS CAL. B0 — Access and change the monitor file. A0 — Read and write error file invoke diagnostic. 80 — Examine (but not change) the monitor. 40 — Default privilege level. {B / O / G} Mname = {value / DELE[TE]} {B / O / G} Pname = {Y[ES] / N[O] / DELE[TE]} XA[CCT] = character string
D[EFAULT]	Changes the defaults recognized by Super. Options: Same as options for the CREATE command.
E[ND]	Causes exit from Super to TEL or to CCI.
F[AST]	Causes a large batch run to be executed in a much shorter time than in normal mode.
L[IST] {[account[,name[,name[,name]]]] / [account[(name[,name[,name])]]]}	Lists the specified contents of all the log-on files: Options: AL[L] PA[SSWORD] XA[CCOUNTING] CA[LL] $BI[LLING] $PR[IVILEGE] Mname Pname $ M P
M[ODIFY] {account,name / account(name)}	Changes the specified fields of an existing log-on record. Options: Same as options for the CREATE command. NONE clears the specific option field in the record to zero.

Table 11. Super Command Summary (cont.)

Command	Description
R[EMOVE] {account,name / account(name)}	Deletes a record from the log-on file. There are no options for this command.
W[ORKSTATION] id	Authorizes a workstation, specifies or change options for a particular workstation, and lists options for a particular workstation or for all workstations. General Options: LW[=ALL] TYPE=type SY[STEM] NS[YSTEM] RP=n GJOB=name MRB NMRB NEM EM LPP=value MLP=value MCP=value MS[T] SL[V] DS[M]=mask X1 N1 X2 N2 RM[T]=nn RW[SN] xxxx DEV devname DD devname Device Options: IN OUT RC[B] value IR[CB] value SU[SBIT] value SR[CB] x LI[ST] x CT[L] NC[TL] DC type SM[D] NS[MD] BI[NARY] NB[INARY] MA[XREC] n MI[NREC] n PR[IV] p KEEP
X id	Deletes the definition of a workstation.

5. USE ACCOUNTING

INTRODUCTION

Since accounting is a matter of installation and corporate preference, the gathering and recording of accounting information is centralized in CP-V. The function of gathering CPU time is centralized in the resident monitor routine T:ACCT. The function of calculating charges and distributing output to users and files is centralized in the shared processor LOGON/LOGOFF. This centralization makes it easy for system programmers to modify the accounting facilities.

The activities of each job, whether initiated on-line or submitted through the batch stream, are accounted for through a series of counts kept in the job information table (JIT) and the assign/merge record associated with each job. These counts record the details of CPU usage, elapsed time, I/O activity, and amount of peripheral use.

At the end of each job, LOGOFF combines these counts with rates from a rate table and calculates total charge units. It then writes an accounting summary record, which may be used for billing or analysis, into the accounting log file, :ACCTLG, which is in the :SYS account. Only jobs with :SYS account numbers are allowed to access the :ACCTLG file.

Deleting the :ACCTLG file has no effect on the operation of the system. The system merely begins a new :ACCTLG file at the termination of the job that did the deleting. Neither deletion nor backup of the :ACCTLG file is automatic. The information contained in the file may or may not be of critical importance for an installation, and it is left to the installation manager to decide how to handle the file.

INSTALLATION ACCOUNTING ROUTINES

In order to provide the installation with the capability of policing the entry of jobs and modifying accounting charges, exits are provided at job and terminal session initiation and termination for the inclusion of supplementary accounting routines to be supplied by the installation.

The installation may supply a batch job or terminal session initiation accounting routine whose DEFed entry point is M:ACINIT and a batch job or terminal session termination accounting routine whose DEFed entry point is M:ACTERM. Briefly, exits are made to these names in the following manner.

M:ACINIT

Assembled as an SREF at:

1. Batch job initiation

2. Terminal session log-on

Calling sequence:

```
BAL, D4    M:ACINIT
```

Input registers:

Register 3 – contains the address of the image of the :USERS record.

Register 5 – contains the address of the JIT.

Output registers:

Register 3 is either zero or nonzero as determined by the installation's M:ACINIT routine. All other registers must remain as they were before the installation's routine was entered.

M:ACTERM

Assembled as an SREF at:

1. Batch job termination

2. Terminal session log-off

Calling sequence:

```
BAL, D4    M:ACTERM
```

Input register:

Register 3 – contains the address of the image of the account record.

Output registers:

Register 3 contains either zero or nonzero as a result of the action of the M:ACTERM routine. Upon return from this routine, all registers except register 3 are expected to remain intact.

In each case, the address of the JIT can be found in location X'4F'. An installation wishing to supplement the standard validation and accounting provided by the system must include ROMs with the above entry names into the appropriate processor load modules during the PASS3 portion of SYSGEN by adding the ROM file names (and account, if needed) to the list of element files on the LOCCT processor commands. The initiation routine, M:ACINIT, is called by both LOGRT and LOGON and must be included in both the CCI and LOGON load modules. The termination routine, M:ACTERM, is called by ACCTSUM and must be included in both LOGON and GHOST1.

Exit will be made at job and terminal session initiation to the installation accounting routine, M:ACINIT. At this point, syntax checking on the control command to enter the system has already been performed. Relevant items in the JIT have already been set up to reflect the environment in which the job is to be executed. The optional extended accounting information, if specified, has been recorded in the assign/merge table and items such as the remaining permanent secondary storage for the job have also been stored in the table. In addition, for batch entries, resource requests (core, tape drives, disk pack spindles) have already been recorded. By pooling the information in JIT, user's :USERS record and assign/merge table, the installation accounting routine can then police the job. If the routine decides that the job is to be denied further access to the system, a zero should be returned in register 3, in which case, the system will output an appropriate message to the user and the job will be aborted.

At job and terminal session termination, exit is made to the installation accounting routine, M:ACTERM. At this point, the system has already prepared an accounting record image. However, before the record is actually appended to the system accounting file, :ACCTLG, the installation accounting routine has the option of modifying the information in the record image. The routine can also determine whether the record should be included in the :ACCTLG accounting file. In the event that the record is not to be written to the file, a zero in register 3 is returned by the routine to the system and the accounting record will be suppressed.

ACCOUNTING LOG FORMAT

Figure 7 shows the format of records in the accounting log file. One record is created for each job or user processed and is retained in the file. The items within a record are defined in Table 12.

ACCOUNTING OUTPUT

The output of accounting information may take either one of two forms. One form consists of a summary of accounting information. The other form consists of the entire accounting record.

For the on-line user, a summary of accounting information is sent to the terminal at the time the user logs off. The format of this information is

CPU m.mmmm CON h:mm INT nn CHG xxxx

where

m.mmmm is CPU time expressed in minutes and ten-thousandths of a minute.

h:mm is console time expressed in hours (h) and minutes (mm).

nn is the number of terminal interactions.

xxxx is total charge units for the on-line session.

The same information may be requested by the user during an on-line session by entry of the TEL STATUS command.

For the batch user, the entire accounting record is written through the M:LL DCB normally assigned to the line printer. The format of this printout is shown in Table 13. The batch user may assign M:LL to a file. However, any errors encountered while attempting to output accounting information through M:LL will force the assignment of M:LL to return to the line printer.

Normally, elapsed time for accounting information is expressed in hours and minutes. The capability exists for the modules TEL, JOBR (CCI), LOGON, and ACCTSUM to express elapsed time in hours, minutes, and seconds. The format is h:mm:ss. Start and end time in the accounting record is converted to seconds from midnight instead of minutes from midnight. This option is controlled by an assembly switch when the modules are assembled.

ACCOUNTING CHARGES

Each user is supplied at log-on time with a charge class value from his log-on record. This value is transferred to the assign/merge table by LOGON and is used by LOGOFF to access the proper rate table in the :RATE file.

:RATE FILE

The :RATE file is a one-record file containing a set of eight rate table pointers and eight rate tables (Figure 8). LOGOFF uses the charge class value for each user to access the associated pointer in the :RATE file which points to the appropriate rate table.

RATES PROCESSOR

The :RATE file is maintained by a specially authorized processor called RATES. The RATES processor, like Super and Control, is accessible from the user account :SYS. If the file :RATE does not exist, RATES creates it using the default value shown in Figure 8.

The RATES processor is called by entering the name of the processor in response to a TEL prompt for a command.

!RATES

-

The :RATE file (if it exists) is automatically loaded into core for modification by RATES commands.

Word	Definition			
0	Account			
2	Name			
5	Extended Accounting			
11	Charge Units			
12	Line Number	Priority	Final Run Status	Job Steps
13	Job Origin	System Version		
14	Start Date			
15	Start Time			
16	End Time			
17	Console Interactions			
18	Finish Date			
19	SYSID		Pack Mounts	Spindles
20	Cards Read		Cards Punched	
21	Processor Pages		User Pages	
22	Diagnostic Pages		Tape Mounts	Tape Drives
23	Tape Accesses			
24	RAD Accesses			
25	Disk Accesses			
26	I/O CALS			
27	Permanent RAD Granules			
28	Permanent Disk Granules			
29	Core Usage			
30	Processor Execution Time			
31	Processor Service Time			
32	Maximum Core Size	Partition	Save Tapes	
33	User Execution Time			
34	User Service Time			
35	Peak Temporary RAD Granules			
36	Peak Temporary Disk Granules			
37	Billing Rate			
38	Accounting Record Status Code	Resource Allocation Values		
⋮				

Figure 7. Structure of Accounting Record

Table 12. Contents of Accounting Record

Name	Description
Account	Account number of the user as specified on the job card or log-on message.
Name	Name of the user as specified on the job card or log-on message.
Extended Accounting	Installation-defined accounting information as specified on the job card or log-on message.
Charge Units	Accumulated charge units calculated for the user through use of the rates table.
Line Number	Line number (Data Set Controller Subchannel) to which the user connected. Line Number is set to X'FF' if entry is for a batch job.
Priority	Priority specified on the job card. Unused if entry is for a terminal session.
Final Run Status	Run status at the completion of the job (an eight bit field). X'00' - Job exited normally. X'01' - Job aborted, illegal trap. X'02' - Job aborted, I/O error. X'04' - Job aborted, limit exceeded. X'08' - Reserved for CHKPT. X'10' - Job aborted, 'X' key-in. X'20' - Last job step errored, 'E' key-in. X'40' - Job aborted, M:xxx. X'80' - Last job step errored, M:ERR.
Job Steps	Total number of job steps if batch; total number of processor operations if on-line.
Job Origin	Origin of batch job. 0 - From local card reader 1 - From on-line terminal 2 - From remote processing
System Version	Version of operating system (from cell X'2B') in EBCDIC.
Start Date	Date at job or terminal session start, where the left halfword is the year and the right halfword is the day. Year is a binary value that records only the last two digits of the year; e.g., 1970 is represented as X'46'. Day is the Julian day of the year represented in binary; e.g., September 14 is represented as X'101'.
Start Time	Time of day at start of job of terminal session in minutes from midnight. The value is expressed in binary.
End Time	Time of day at end of job or terminal session. Expressed in the same format as start time.
Console Interactions	Number of interactions during the course of a terminal session (zero for batch).

Table 12. Contents of Accounting Record (cont.)

Name	Description
Finish Date	Date at job or terminal session finish. The format is the same as for Start Date.
Sysid	ID assigned to user job or session.
Pack Mounts	Number of disk packs mounted.
Spindles	Maximum number of disk pack spindles allocated to batch or available to on-line.
Cards Read	Number of cards read, including the job card and any EOD cards, but not FIN cards.
Cards Punched	Number of cards punched, including ID card, JOB card, BIN cards and EOD cards, but not blank cards inserted by the punch symbiont between jobs. If no punched output is produced by the job, the ID and JOB cards are suppressed and the punched card count is reduced to zero.
Processor Pages	Number of pages of printed output generated by shared processors, plus two ID pages at the beginning of the job and the accounting page at the end.
User Pages	The number of pages of printed output generated by user programs only.
Diagnostic Pages	The number of pages of all output to a symbiont file through the M:DO DCB, including core dump snaps and debug output. However, output is not counted if it goes to a user file, even though it goes via M:DO.
Tape Mounts	Number of tapes mounted unless premounted by the operator.
Tape Drives	Maximum number of tape drives allocated to batch or available to on-line.
Tape Accesses RAD Accesses Disk Accesses	Number of read, write, and file positioning accesses on the specified device, but not seek accesses since these are considered part of a read or write. A chargeable access is actually a request to a queue.
I/O CALs	Number of CAL1, 1 operations performed.
Permanent RAD Granules	Net change in accumulated RAD storage. This is a signed binary value.
Permanent Disk Granules	Net change in accumulated public disk pack storage. This is a signed binary value.
Core Usage	Product of CPU time times core size in pages (ticks[†] x pages). Includes all core usage by job.
Processor Execution Time	CPU time spent in shared processors in the slave mode, expressed in ticks.[†]
Maximum Core Size	Peak value of core reached, expressed as the number of pages. Does not include shared processors or context.
Partition	Partition number under which the job ran (zero if terminal session).
Save Tapes	Number of save tapes used.
User Execution Time	CPU time spent in other than shared processors expressed in ticks.[†]
User Service Time	Monitor service time spent for other than shared processors expressed in ticks.[†]
Peak Temporary RAD Granules	Peak value of temporary RAD granules used.

Table 12. Contents of Accounting Record (cont.)

Name	Description
Peak Temporary Disk Granules	Peak value of temporary public disk pack granules used.
Billing Rate	Charge class used for accounting for this user. The value is obtained from the user's log-on record and is in the range 0 to 7.
Accounting Record Status Code	Status under which the accounting record was generated. X'00' – Accounting record was created under normal conditions. X'01' – Accounting record was created during recovery but the accounting record is correct. X'02' – Accounting record was created under abnormal conditions (assign/merge read error) and contains erroneous information for Start Date, Start Time, Billing Rate, Permanent RAD Granules, and Permanent Disk Granules. X'03' – Accounting record was created under abnormal conditions (assign/merge read error) during system recovery and contains erroneous information for Start Date, Start Time, Billing Rate, Permanent RAD Granules, and Permanent Disk Granules.
Resource Allocation Values	Values of resources at time of log-off. Values are in one byte fields and appear in the same order as the resources specified in the system resource limit table.

†One tick equals two milliseconds.

Table 13. Accounting Printout for Batch Jobs

Printed Format			Explanation
(Time and Date)			
ELAPSED JOB TIME		hh:mm	Clock time in hours and minutes for job or terminal session.
PARTITION NUMBER			Partition number under which the job ran.
CHARGE UNITS		xxxxxxxx	Total charge units.
TOTAL CPU TIME		x.xxxx	Sum of all execution time (in minutes).
	PROCESSOR EXECUTION TIME	x.xxxx	Shared processor execution time (e.g., FORTRAN) (in minutes).
	PROCESSOR SERVICE TIME	x.xxxx	Monitor time for CALs issued by shared processors (in minutes).
	USER EXECUTION TIME	x.xxxx	User program execution (in minutes).
	USER SERVICE TIME	x.xxxx	Monitor time for user issued CALs (in minutes).
CARDS:	CARDS READ	xxxx	Number of cards read.
	CARDS PUNCHED	xxxx	Number of cards punched.

Table 13. Accounting Printout for Batch Jobs (cont.)

Printed Format			Explanation
PAGES:	PROCESSOR PAGES	xxxx	Number of pages printed by shared processors.
	USER PAGES	xxxx	Number of pages printed by user program.
	DIAGNOSTIC PAGES	xxxx	Number of pages printed through M:DO.
TAPES:	TAPES MOUNTED	xx	Number of tapes mounted.
	DRIVES ALLOCATED	xx	Number of tape drives allocated.
	SAVE TAPES USED	xx	Number of save tapes used.
PACKS:	PACKS MOUNTED	xx	Number of disk packs mounted.
	SPINDLES ALLOCATED	xx	Number of disk spindles allocated.
CORE:	PEAK CORE (PAGES)	xxx	Maximum number of core pages used at any one time. Does not include shared processors.
	PAGE * MINUTES	xxxxxx	Amount of core time used. Includes swappable core usage (data and context).
I/O:	OPERATIONS	xxxxx	Number of physical I/O actions except terminal and swap I/O.
	CALS	xxxxxx	Number of CAL,1 operations.
FILE SPACE			
PEAK RAD TEMPORARY		xxxx	Peak value of temporary RAD granules used.
NET RAD PERMANENT		xxxx	Net change in accumulated RAD storage (in granules).
AVAILABLE RAD PERMANENT		xxxx	Amount of RAD space available for permanent storage (in granules).
PEAK DISK TEMPORARY		xxxx	Peak value of temporary public disk pack granules used.
NET DISK PERMANENT		xxxx	Net change in accumulated public disk pack storage (in granules).
AVAILABLE DISK PERMANENT		xxxx	Amount of public disk pack space available for per permanent storage (in granules).

RATES COMMANDS

There are three commands in the RATES processor command language. They are BUILD, PRINT, and END.

RATES prompts for a command by typing a single dash. It prompts for each charge rate table entry for the BUILD command by typing a charge siding followed by an equal sign.

RATES looks only at the first and last characters of a command. The first character identifies the command; the last character identifies the charge rate table the user wishes to manipulate (for BUILD and PRINT). Thus, RATES accepts neither leading nor trailing blanks in command input. Only the last digit of a multidigit number will be used to identify a rate table.

BUILD The BUILD command modifies the charge unit values in the specified charge rate table. The format of the command is

-B[UILD] table

CPU TIME= [value]

CPU TIME * CORE SIZE= [value]

TERMINAL INTERACTIONS= [value]

I/O CALS= [value]

CONSOLE MINUTES= [value]

Charge class 0 pointer	0	8	Charge class pointers point to charge rate tables
Charge class 1 pointer	1	16	
Charge class 2 pointer	2	24	
Charge class 3 pointer	3	32	
Charge class 4 pointer	4	40	
Charge class 5 pointer	5	48	
Charge class 6 pointer	6	56	
Charge class 7 pointer	7	64	
CPU time	8	6	Charge rate table 0 (default table for batch jobs)
CPU time x core size	9	1	
Terminal interactions	10	0	
I/O CALs	11	1000	
Console minutes	12	0	
Tapes and packs mounted	13	100000	
Page-date storage	14	10000	
Peripheral I/O cards and pages	15	2000	
	16	6	Charge rate table 1 (default table for on-line jobs)
.	17	1	
.	18	1000	
.	19	1000	
	20	200	
	21	100000	
	22	10000	
	23	2000	
		. . .	Charge rate tables 2–6
	64	6	Charge rate table 7
	65	1	
	66	1000	
	67	1000	
	68	200	
	69	100000	
	70	10000	
	71	2000	

Figure 8. Structure of :RATE File

TAPES AND PACKS MOUNTED= [value]

PAGE - DATE STORAGE= [value][†]

PERIPHERAL I/O CARDS + PAGES= [value]

where

table specifies one of the eight charge rate tables and has a range $0 \leq$ table ≤ 7.

value specifies the units to be charged for a particular item and has a range $0 \leq$ value ≤ 99999999. If the user does not enter a value before entering a carriage return character, RATES will not alter the stored value for that item.

RATES allows the user to make two input errors before requesting the user to retype the command. It stores all legal values it receives prior to the error.

Example:

Assume that the user wants to change some of the charge unit values in charge rate table 3. Specifically, he wants to change the charge units for terminal interactions and I/O CALs to 5000 and 15,000 respectively.

```
- BUILD 3 (RET)

CPU TIME= (RET)

CPU TIME * CORE SIZE= (RET)

TERMINAL INTERACTIONS= 5000 (RET)

I/O CALS= 15000 (RET)

CONSOLE MINUTES= (RET)

TAPES AND PACKS MOUNTED= (RET)

PAGE - DATE STORAGE= (RET)†

PERIPHERAL I/O CARDS + PAGES= (RET)

-
```

PRINT The PRINT command prints the contents of the specified charge rate table. The format of the command is

P[RINT] table

where

table specifies the charge rate table to be printed and has a range $0 \leq$ table ≤ 7.

[†] Although rate table entries are made for this item, charges to it are not recorded.

Example:

Assume the user wants to print the contents of charge rate table 3.

```
- P3 (RET)

CPU TIME= 6

CPU TIME * CORE SIZE= 1

TERMINAL INTERACTIONS = 5000

I/O CALS = 15000

CONSOLE MINUTES = 200

TAPES AND PACKS MOUNTED = 100000

PAGE - DATE STORAGE = 10000†

PERIPHERAL I/O CARDS + PAGES = 2000

-
```

END The END command causes RATES to write the :RATE file on disk storage and to exit to TEL. The format of the command is

E[ND]

Example:

```
- END (RET)
!
```

BREAK CONTROL

If the user depresses the BREAK key anytime after he receives the first prompt for command input and before he enters the END command, RATES stops processing the current command and returns to the command level. If the user depresses the BREAK key after entering the END command, RATES ignores the break and processes the END command. At all other times, RATES returns to TEL when the user depresses the BREAK key.

RATES ERROR MESSAGES

RATES returns one or more error messages to the user's terminal if the user incorrectly enters a command. These messages are listed in Table 14.

RATES COMMAND SUMMARY

Table 15 summarizes the RATES commands. The left-hand column lists the command format and the right-hand column defines the command values.

Table 14. RATES Error Messages

Message	Description
ILLEGAL COMMAND	A command other than BUILD, PRINT, or END was entered in response to a prompt for command input. Retype the command.
ILLEGAL NUMBER	A value entered in response to a BUILD command prompt was nonnumeric or exceeded eight digits. Retype the command if there is no other message.
LAST CHARACTER MUST BE 0-7	The last character of a BUILD or PRINT command was not a digit in the range 0-7. Retype the command.
:RATE FILE DID NOT EXIST, BUT HAS BEEN CREATED	RATES created the :RATE file with the default values given in Figure 8. (This is not an error message.)
RETYPE COMMAND	A value entered in response to a BUILD command was incorrect. Retype the command. (RATES stores legal values that were input prior to the incorrect value.)
SORRY, YOU ARE NOT ALLOWED TO ACCESS THE :RATE FILE	An attempt was made to access the RATES processor by a user who is not logged on with the name LBE under account :SYS.
UNEXPECTED I/O ERROR NO UPDATING TOOK PLACE	RATES encountered an unexpected I/O error while opening, reading, writing, or closing the :RATE file. Call the RATES processor again.

Table 15. RATES Command Summary

Command	Description
-B[UILD] table CPU TIME [value] CPU TIME * CORE SIZE [value] TERMINAL INTERACTIONS [value] I/O CALS [value] CONSOLE MINUTES [value] TAPES AND PACKS MOUNTED= [value]	Modifies the charge unit values in the specified charge rate table. All values must be in the range $0 \leq \text{value} \leq 99999999$. If no value is entered prior to a carriage return character, the currently stored value is saved.
PAGE - DATE STORAGE [value] PERIPHERAL I/O CARDS · PAGES [value]	Although rate table entries are made for this item, charges to it are not recorded.
E[ND]	Writes the updated :RATE file and returns control to TEL.
P[RINT] table	Prints the contents of the specified charge rate table.

6. SYSTEM PERFORMANCE CONTROL

INTRODUCTION

CP-V has a comprehensive set of performance measurement and system control facilities. These facilities allow the system manager to determine how the system is performing and to adjust critical operational parameters to achieve better performance.

The three processors that provide these facilities are briefly described below and will be discussed in detail in later sections of this chapter.

1. The Control processor allows the system manager to display and modify certain system parameters so that the system can be "tuned" to meet the needs of the particular installation.

2. The STATS processor allows the system manager to display current statistical information about the system and to collect a series of "snapshots" of statistical information that provide a history of system operation.

3. The Summary processor enables the system manager to obtain various types of statistical reports using the information in the history files created by STATS.

DISPLAY ITEMS

Before discussing the three processors, certain groups of display items should be described and listed. Several of these display items are utilized in both Control and STATS operations. There are other groups of display items that are applicable to one processor only. Such items will be discussed in the section about the particular processor.

The display items to be described here are control parameters, current values, and partition attributes.

A control parameter is a system parameter that can be modified to tune the system (see Table 16). For example, the maximum number of on-line users is a control parameter. Changing its value may change average response time as well as other performance characteristics.

A current value is a system parameter that reflects the current state of the system and cannot be modified (see Table 17). For example, the current number of 7-track tape drives allocated to batch is a current value. A display of current values renders a crude estimate of current utilization of system resources.

Batch partition attributes are system parameters and current values that pertain only to a given partition. (Partitions are described in this chapter in the section "Partition Display and Definition Commands".) Certain attributes define the conditions that a job must satisfy to run under a given partition (see Table 18). For example, the maximum amount of time a job may execute before termination is a partition attribute. Partition attributes can also be thought of as system resources that the partition can offer a job. If a job requires greater or fewer resources than defined for a given partition, the job may not be run in that partition. Certain attributes reflect the current state of a partition and may not be altered. For example, the account number of the user currently executing in a given partition is such an attribute.

CONTROL

The Control processor provides control over system performance. There are a number of performance measurements built directly into the system. Commands of the Control processor enable the system manager to display these measurements and to "tune" the system as needed by setting new values for the parameters that control system performance.

A user privilege of 80 is required for displaying information. A privilege of B0 is required for modifying the system parameter limits found in Table 14. A privilege of C0 is required for modifying partition attributes.

Control may be run as a batch, ghost, or on-line job. Three DCBs are used for input and output (M:SI, M:LO, and M:DO). If run as a batch or on-line job, Control inputs may be stored in a file which must be ASSIGNed (SET) to the M:SI DCB. Displays and reports may be output to a file by reassignment of the M:LO DCB. Operator instructions and error messages may be output to a file by reassigning the M:DO DCB. A summary of default and possible DCB assignments is given in Table 19.

In the batch mode, Control is called with the !CONTROL card. Control commands are placed on cards, one per card, anywhere within the first 37 characters of the card.

Control may be initiated as a ghost job by the operator key-in !GJOB CONTROL. During initiation, if the standard DCBs through which input/output occur have not previously been assigned, Control sets the three DCBs through which communication occurs (M:SI, M:LO, and M:DO) as follows: M:SI = OC, M:DO = OC, and M:LO = OC. Therefore, all commands are input through the operator's console and output occurs through the operator's console.

The Control processor is called on-line by entering CONTROL as a TEL command. Control responds by typing CONTROL HERE and then prompts for a command using a dash (–) as a prompt character.

Table 16. Control Parameters

Control Name	System DEF	Description	Unit	Minimum Value	Maximum Value
		User Maximums			
BUM	S:BUAIS	Maximum number of concurrent batch users.	users	0	SMUIS-① OUM- S:GUAIS
OUM	S:OUAIS	Maximum number of on-line users allowed in the system.	users	0	SMUIS- BUM- S:GUAIS
		Execution Control			
BB	SL:BB	Batch bias. Zero indicates batch compute-bound tasks have less priority than on-line compute-bound tasks. Nonzero indicates batch and on-line compute-bound tasks have equal priority.	-	0	100
PI	SL:PI	Priority increment (in units of 1/256) to be used for increasing the priority of jobs bypassed by the multi-batch scheduler.	1/256	0	255
QUAN	SL:QUAN	Time-slice by which compute-bound users are shared.	msecs	QMIN	5,000
QMIN	SL:QMIN	Amount of uninterrupted compute time guaranteed a user after selection.②	msecs	0	10,000
SQUAN	SL:SQUAN	Amount of time a user is guaranteed core residency before swap out.②	msecs	0	10,000
BPRIO	SL:BPRIO	Batch base execution priority.	-	X'C0'③	X'FF'③
OPRIO	SL:OPRIO	On-line base execution priority	-	X'C0'③	X'FF'③
GPRIO	SL:GPRIO	Ghost base execution priority	-	X'C0'③	X'FF'③
		I/O Control			
BXMF	SL:BXMF	Maximum number of concurrent I/O functions per batch user. If exceeded, the user is blocked.	numeric	BIMF	255
BIMF	SL:BIMF	Lower threshold value for number of concurrent I/O functions per batch user. When number of functions drops below this value, the user is unblocked.	numeric	0	BXMF
OXMF	SL:OXMF	Maximum number of concurrent I/O functions per on-line user. If exceeded, the user is blocked.	numeric	OIMF	255
OIMF	SL:OIMF	Lower threshold value for number of concurrent I/O functions per on-line user. When number of functions drops below this value, the user is unblocked.	numeric	0	OXMF
TB	SL:TB	Number of characters at which to block terminal output.④	chars.	UB	256
UB	SL:UB	Number of characters at which to unblock terminal output.⑤	chars.	1	TB

Table 16. Control Parameters (cont.)

Control Name	System DEF	Description	Unit	Minimum Value	Maximum Value
		I/O Control (cont.)			
ONCB	SL:ONCB	Maximum number of COC buffers allowed per user.	buffers	2	255
OLTO	SL:OLTO	Log-on time out.	minutes	1	$2^{16}-1$
OITO	SL:OITO	Terminal input time out.	minutes	1	$2^{16}-1$
RAM	SL:RAMR	Maximum number of concurrent read-ahead operations.	numeric	0	RASIZE
RATO	SL:RATOR	Time after which a read-ahead operation will be aborted.	msecs	0	32,767
		Exit Control Processing Limits			
ETIME	SL:ETIME	Maximum exit control execution time allowed.	seconds	0	$2^{31}-1$
ELO	SL:ELO	Additional number of processor pages that may be listed through the M:LO DCB after exceeding the batch or on-line limit.	pages	0	32,767
EPO	SL:EPO	Additional number of object records that may be output after exceeding the batch or on-line limit.	records	0	32,767
EDO	SL:EDO	Additional number of pages of diagnostics that may be listed after exceeding the batch or on-line limit.	pages	0	32,767
EUO	SL:EUO	Additional number of pages of user output that may be listed after exceeding the batch or on-line limit.	pages	0	32,767
ETS	SL:ETS	Additional amount of temporary disk storage allowed after exceeding the batch or on-line limit.	granules	0	65,535
EPS	SL:EPS	Additional amount of permanent disk storage allowed after exceeding the batch or on-line limit.	granules	0	65,535
		Resource Limit Control			
Tres⑥	SH:RTOT	Total resource available for all jobs.	⑦	0	System Capability
BTres⑥	SH:RBSUM	Total resource available for all batch jobs.	⑦	0	SH:RTOT
OTres⑥	SH:ROSUM	Total resource available for all on-line jobs.	⑦	0	SH:RTOT
GTres⑥	SH:RGSUM	Total resource available for all ghost jobs.	⑦	0	SH:RTOT
BCres⑥	SH:RBCU	Current value of a resouce allocated to all batch jobs.	⑦	-	-
OCres⑥	SH:ROCU	Current value of a resource allocated to all on-line jobs.	⑦	-	-
GCres⑥	SH:RGCU	Current value of a resource allocated to all ghost jobs.	⑦	-	-

Table 16. Control Parameters (cont.)

Control Name	System DEF	Description	Unit	Minimum Value	Maximum Value
		Resource Limit Control (cont.)			
BMres⑥	SB:RBMX	Maximum value of a resource that can be requested by a batch job.	⑦	0	SH:RBSUM
OMres⑥	SB:ROMX	Maximum value of a resource that can be requested by an on-line job. This value is not used at present.	⑦	0	SH:ROSUM
GMres⑥	SB:RGMX	Maximum value of a resource that can be requested by a ghost job. This value is not used at present.	⑦	0	SH:RGSUM
BDres⑥	SB:RBDF	Default value of a resource that is allocated to a batch job.	⑦	0	SB:RBMX
ODres⑥	SB:RODF	Default value of a resource that is allocated to an on-line job.	⑦	0	SB:ROMX
GDres⑥	SB:RGDF	Default value of a resource that is allocated to a ghost job.	⑦	0	SB:RGMX
		Service Limit Control			
BMserv⑧	SL:BMX	Maximum value of a service that can be requested by a batch job.	⑨	0	32,767
OMserv⑧	SL:OMX	Maximum value of a service that can be requested by an on-line job.	⑨	0	32,767
GMserv⑧	SL:GMX	Maximum value of a service that can be requested by a ghost job.	⑨	0	32,767
BDserv⑧	SL:BDF	Default value of a service that can be assigned to a batch job.	⑨	0	SL:BMX
ODserv⑧	SL:ODF	Default value of a service that can be assigned to an on-line job.	⑨	0	SL:OMX
GDserv⑧	SL:GDF	Default value of a service that can be assigned to a ghost job.	⑨	0	SL:GMX
		Symbiont and Special Feature Limit Control			
BDfa⑩	S:SYMDB	Default authorization of a symbiont device or a special feature for a batch job.	-	0⑪	1⑪
ODfa⑩	S:SYMDO	Default authorization of a symbiont device or a special feature for an on-line job.	-	0⑪	1⑪
GDfa⑩	S:SYMDG	Default authorization of a symbiont device or a special feature for a ghost job.	-	0⑪	1⑪

① SMUIS is the maximum number of users (sum of SYSGEN parameters MAXB, MAXO, and MAXG).

② If this value is greater than QUAN, QUAN is assumed by the system.

③ Execution priorities are inversely related to numeric value. That is, X'FF' is the lowest execution priority and X'C0' is the highest execution priority. X'BF' through X'00' are reserved for real-time processing.

Table 16. Control Parameters (cont.)

④ TB is the number of characters at which the user is swapped out to the swapping disk.

⑤ UB is the number of characters at which the user is swapped back into core memory.

⑥ This is a generalized format, where "res" is replaced by one of the two-character resource names defined on the :RES command at SYSGEN. Standard, SYSGEN default, resource names are

CO	-	core	7T	-	7-track tapes
9T	-	9-track tapes	SP	-	disk pack spindles

⑦ Unit depends on the particular resource. Units for standard resources are "tape drives" for tapes, "spindles" for disk packs, and "K words" for core.

⑧ This is a generalized format where "serv" is replaced by one of the two- or four-character service names. The service names are listed in Table 6.

⑨ Unit depends on the particular service. The units are indicated in the descriptions of the services in Table 6.

⑩ This is a generalized format where "fa" may be replaced by the 2-character name of a symbiont device (i.e., the device type) or by the 2-character identifier of a special feature defined at SYSGEN by the :FAUTH command (e.g., EQ for the enqueue/dequeue feature).

⑪ The value 0 indicates that the symbiont device is not authorized. The value 1 indicates that the symbiont device is authorized.

Table 17. Current System Values

Control Name	System DEF	Description	Units
BCCO	SH:RBCU	Current core size allocated to concurrent batch users (whether occupying memory or swapped out).	K words
BC7T	SH:RBCU	Current number of 7-track tape drives allocated to all batch users.	7T drives
BC9T	SH:RBCU	Current number of 9-track tape drives allocated to all batch users.	9T drives
BCSP	SH:RBCU	Current number of disk pack spindles allocated to all batch users.	spindles
OC7T	SH:ROCU	Current number of 7-track tape drives allocated to all on-line users.	7T drives
OC9T	SH:ROCU	Current number of 9-track tape drives allocated to all on-line users.	9T drives
OCSP	SH:ROCU	Current number of disk pack spindles allocated to all on-line users.	spindles
UC	S:CUIS	Current number of users in the system.	users
BUC	S:BUIS	Current number of batch users in the system.	users
OUC	S:OUIS	Current number of on-line users in the system.	users
GUC	S:GUIS	Current number of ghost users in the system.	users
GUM	S:GUAIS	Maximum number of ghost users allowed in the system.	users
UM	SMUIS	Maximum number of users allowed in the system.	users
GCCO	SH:RGCU	Current core size allocated to ghost users (whether occupying core memory or swapped out).	K core
GC7T	SH:RGCU	Current number of 7-track tape drives allocated to all ghost users.	7T drives
GC9T	SH:RGCU	Current number of 9-track tape drives allocated to all ghost users.	9T drives
GCSP	SH:RGCU	Current number of disk pack spindles allocated to all ghost jobs.	spindles

Table 18. Multi-Batch Partition Attributes

Control Name	System DEF	Display Only	Description	Units	Minimum Value	Maximum Value
TIME①	PLH:TL		Minimum job execution time for jobs to be selected for this partition.	minutes	0	PLH:TU
TIME①	PLH:TU		Maximum job execution time for jobs to be selected for this partition.	minutes	PLH:TL	32,767
QUAN	PLH:QN		Time-slice for this partition.	msecs.	0	5,000
ACCT②	PLD:ACT	X	Account number of job currently executing in partition.	EBCDIC	-	-
CUR②	PLH:CUR	X	Number of jobs that have been run under current definition of this partition. ③	jobs	0	32,767
TOL②	PLH:TOL	X	Total number of jobs run under this partition since system startup. ③	jobs	0	32,767
HOLD	PLH:FLG④		Core residency control flag. 1 → hold job in core. 0 → allow swap out.	boolean	0 or 'NO'	1 or 'YES'
LOCK②	PLH:FLG		Partition selection lock flag. 0 → select jobs for this partition. 1 → lock partition from further selection.	boolean	0 or 'NO'	1 or 'YES'
USER	PLB:USR	X	ID number of user currently executing in partition.	hexadecimal	0	X'FF'
SP①	PLB:MAX⑤		Maximum number of spindles that may be used for selection under this partition.	spindles	0	SB:RBMX
7T①	PLB:MAX		Maximum number of 7-track drives that may be used for selection under this partition.	7T drives	0	SB:RBMX
9T①	PLB:MAX		Maximum number of 9-track drives that may be used for selection under this partition.	9T drives	0	SB:RBMX
CO①	PLB:MAX		Maximum size of core that may be used for selection under this partition.	K words	0	SB:RBMAX
SP①	PLB:MIN		Minimum number of spindles that may be used for selection under this partition.	spindles	0	PLB:MAX
7T①	PLB:MIN		Minimum number of 7-track drives that may be used for selection under this partition.	7T drives	0	PLB:MAX
9T①	PLB:MIN		Minimum number of 9-track drives that may be used for selection under this partition.	9T drives	0	PLB:MAX
CO①	PLB:MIN		Minimum size of core that may be used for selection under this partition.	K words	0	PLB:MAX

① TIME, 7T, 9T, SP, and CORE specify upper and lower resource limits for the partition and are displayed as a range when the resource name is referenced.

② ACCT, CUR, TOL USER may be displayed but not modified.

③ Not including the currently executing job.

④ Bit 0 of PLH:FLG is the HOLD flag and bit 15 is the LOCK flag.

⑤ PLB:MAX and PLB:MIN are sets of parallel byte tables.

Table 19. Control DCB Usage

DCB	On-line		Batch		Ghost	
	Default	Assigned	Default	Assigned	Default	Assigned
M:SI	User Console (UC)	File	Card Reader (CR)	File, 7T, 9T, or Labeled Tape (LT)	Operator Console (OC)	-
M:LO	User Console (UC)	File, 7T, 9T, Labeled Tape, or Line Printer (LP)†	Line Printer (LP)†	File, 7T, 9T, or Labeled Tape (LT)	Operator Console (OC)	-
M:DO	User Console (UC)	File	Line Printer (LP)	File, 7T, 9T, or Labeled Tape (LT)	Operator Console (OC)	-

†If control display output through the M:LO DCB is directed to the line printer, when the user ends his control session (with an END command), a super-close will be effected, closing the cooperative and outputting the display.

Example:

> !CONTROL
>
> CONTROL HERE
>
> -

All examples of Control commands are written for the on-line mode.

CONTROL PROCESSOR COMMANDS

Commands provided by the Control processor fall into four categories. They are

1. System Parameter Display Commands

   ADD

   DROP

   LIST

   CONTROL

   CONTROL!

   name

2. Control Parameter Assignment Command

   name = value

3. Partition Display and Definition Commands

   ADD

   DROP

   PARTITION

   DISPLAY n

   n attribute

   n attribute = value

   STORE

   CLEAR

   BREAK (i.e.,)

   END

4. Miscellaneous Commands

   QUIT

   PROCEED

   BREAK (i.e.,)

   END

   TIME

The ADD, DROP, BREAK, and END commands appear twice in the list because there are two command levels in the Control processor and the functions of these two commands depend upon the command level at which they are given.

The two command levels are the control command level and the PARTITION sub-command level. The control command level permits execution of all system parameter display commands, the control parameter assignment command, the miscellaneous commands, and the PARTITION command. Initial entry into the Control processor is at this level. The PARTITION command invokes a sub-command level which permits execution of only the partition display and definition commands.

The Control processor prompts for commands at the control command level with a dash (-) and prompts for commands at the subcommand level with a 'greater than' (>) character. The ADD and DROP limited subcommand level prompts with a period (.).

For all Control commands, only the first two letters of the command are significant. For example, the following are equivalent:

-ADD (RET)

-AD (RET)

-ADXYZ (RET)

-ADD ITEMS TO PRINT LIST (RET)

The commands will be discussed in the order in which they were listed above.

SYSTEM PARAMETER DISPLAY COMMANDS

ADD This command turns on the print flags for the specified items. The format of the command is

-AD[D]

.item

.item

.

.

.

.item

where item is one of the control names or current values listed in Tables 16 and 17. The command is terminated by entering a (RET) alone after a prompt for input. If 'ALL' is specified, all print flags will be turned on.

In the following example, the print flags for two control parameters, QMIN and OM7T, are turned on:

-ADD (RET)

.QMIN (RET)

.OM7T (RET)

. (RET)

-

The ADD command enables the user to select a group of parameters to be displayed whenever the display command CONTROL is given. (Other display commands allow the user to display all of the parameters or a single specified parameter.)

DROP The DROP command turns off the print flag for the specified items. The format of the command is

-DR[OP]

.item

.item

.

.

.

.item

where item is one of the control names or current values listed in Tables 16 and 17. If 'ALL' is specified, all print flags will be turned off. The command is terminated by entering a (RET) alone after the prompt for input.

In the following example, the print flags for two control parameters, QMIN and OM7T, are turned off:

-DROP (RET)

.QMIN (RET)

.OM7T (RET)

. (RET)

-

LIST This command lists the names of all items (from the group in Tables 16 and 17) with their print flags on. It can be used to make certain that a series of ADD and DROP commands had the desired effect. The format of the command is

LI[ST]

Example:

-DROP (RET)

.ALL (RET)

. (RET)

-LIST (RET)

-AD (RET)

.QUAN (RET)

.TB (RET)

-.UB (RET)

-.BCSP (RET)

-. (RET)

-DROP

-.TB (RET)

-.UB (RET)

-. (RET)

-LIST (RET)

QUAN

BCSP

-

In this example, the first LIST command lists no items since the print flags for all items were turned off by the DROP command. The second LIST command lists QUAN and BCSP because their print flags were turned on by the ADD command but not turned off by the subsequent DROP command.

CONTROL This command displays all control parameters and current values with their print flags on. (The control parameters and current values are listed in Tables 16 and 17.) The format of the command is

CO[NTROL]

Example:

-DROP (RET)

-.ALL (RET)

-. (RET)

-ADD (RET)

-.OMCO (RET)

-.OUM (RET)

-. (RET)

-CONTROL (RET)

MAX ONLINE JOB RSRCE MCO = 32

MAX # ON-LINE USERS (OUM) = 32

Due to the carriage width limitation of most terminal devices, only four decimal digits of the parameter may be printed with the parameter description. If more decimal digits are required, four asterisks will be printed after the equal sign to indicate overflow. The parameter value will then be automatically displayed as a single parameter display. Currently, control parameters do not require more than four decimal digits, but new values may be added in the future.

CONTROL! This command displays all control parameters and current values whether or not their print flags are set. (The control parameters and current values are listed in Tables 16 and 17.) The format of the command is

CO[NTROL]!

Only the first two characters and the exclamation point are necessary.

The display prints out in the format shown in Figure 9.

name Single control parameters and current values may be displayed by entering their control name as a command. (The control parameters are listed in Table 16 and the current values are listed in Table 17.) All characters of the name must be entered.

Example:

-BTCO (RET)

BTCO = 63

CONTROL PARAMETER ASSIGNMENT COMMAND

name = number The value of a specified control parameter can be set by this command. The format of the command is

name = number

where

name is one of the control parameter names in Table 16. (The current values listed in Table 17 can not be modified.)

number is an unsigned integer in the range for the name.

All characters in the name must be entered. Note that no blanks are allowed.

Example:

-QUAN = 450 (RET)

-QUAN (RET)

QUAN = 450

In this example, the value of the control parameter QUAN is changed to 450 and displayed.

```
MAX #USERS IN SYS        (UM)=    41   CURRENT # USER IN SYS    (UC)=    3
MAX # BATCH USERS       (BUM)=     1   CURRENT # BATCH USERS   (BUC)=    0
MAX # ØNLINE USERS      (ØUM)=    15   CURRENT # ØNLINE USERS  (ØUC)=    1
MAX # GHØST JØBS        (GUM)=    15   CURRENT # GHØST JØBS    (GUC)=    2
BATCH BIAS               (BB)=    50   MSEC ØNLINE QUANTUM    (QUAN)=  300
PRIØRITY INCREMENT       (PI)=     0   ØNLINE MINIMUM QUAN    (QMIN)=   10
MAX BATCH I/Ø CØUNT    (BXMF)=     9   MAX ØNLINE I/Ø CØUNT   (ØXMF)=    9
MIN BATCH I/Ø CØUNT    (BIMF)=     1   MIN ØNLINE I/Ø CØUNT   (ØIMF)=    1
ELIM DEFAULT - PSTØRE   (EPS)=   100   MSEC CØRE-RESIDENCY   (SQUAN)=  300
ELIM DEFAULT - TIME   (ETIME)=    30   LØG-ØN TIME-ØUT        (ØLTØ)=   15
ELIM DEFAULT - LØ       (ELØ)=    10   TERM INPUT TIME-ØUT    (ØITØ)=   30
ELIM DEFAULT - PØ       (EPØ)=    50   # CHARS TERMINAL BLØCK   (TB)=  200
ELIM DEFAULT - DØ       (EDØ)=    10   # CHARS TERM UNBLØCK     (UB)=   14
ELIM DEFAULT - UØ       (EUØ)=    10   MAX # CØC BUFS/USER    (ØNCB)=   15
ELIM DEFAULT - TSTØRE   (ETS)=    64   BASE EXEC PRIØ-BATCH  (BPRIØ)=  254
BASE EXEC PRIØ-ØNLINE(ØPRIØ)=    254   BASE EXEC PRIØ-GHØST  (GPRIØ)=  254
MAXIMUM # READ-AHEADS   (RAM)=     0   READ-AHEAD TIME-ØUTS   (RATØ)=   46
MAX # AIR ENTRIES      (AIRM)=     0   AIR TIME-ØUT          (AIRTØ)= 3840
SYSTEM TØTAL RESØURCE CØ      =32767   SYSTEM TØTAL RESØURCE 9T     =    4
SYSTEM TØTAL RESØURCE 7T      =    1   SYSTEM TØTAL RESØURCE SP     =    3
SYSTEM TØTAL RESØURCE BT      =    4   TØTAL *GHØST*  RSRCE  TCØ    =32767
TØTAL *GHØST*  RSRCE  T9T     =    4   TØTAL *GHØST*  RSRCE  T7T    =    1
TØTAL *GHØST*  RSRCE  TSP     =    3   TØTAL *GHØST*  RSRCE  TBT    =    4
CURNT *GHØST*  RSRCE  CCØ     =    0   CURNT *GHØST*  RSRCE  C9T    =    0
CURNT *GHØST*  RSRCE  C7T     =    0   CURNT *GHØST*  RSRCE  CSP    =    0
CURNT *GHØST*  RSRCE  CBT     =    0   MAX GHØST  JØB RSRCE  MCØ    =  255
MAX GHØST  JØB RSRCE  M9T     =    4   MAX GHØST  JØB RSRCE  M7T    =    1
MAX GHØST  JØB RSRCE  MSP     =    3   MAX GHØST  JØB RSRCE  MBT    =    4
DFT GHØST  JØB RSRCE  DCØ     =  255   DFT GHØST  JØB RSRCE  D9T    =    2
DFT GHØST  JØB RSRCE  D7T     =    1   DFT GHØST  JØB RSRCE  DSP    =    3
DFT GHØST  JØB RSRCE  DBT     =    2   TØTAL *ØNLINE* RSRCE  TCØ    =  200
TØTAL *ØNLINE* RSRCE  T9T     =    3   TØTAL *ØNLINE* RSRCE  T7T    =    1
TØTAL *ØNLINE* RSRCE  TSP     =    3   TØTAL *ØNLINE* RSRCE  TBT    =    3
CURNT *ØNLINE* RSRCE  CCØ     =    0   CURNT *ØNLINE* RSRCE  C9T    =    0
CURNT *ØNLINE* RSRCE  C7T     =    0   CURNT *ØNLINE* RSRCE  CSP    =    0
CURNT *ØNLINE* RSRCE  CBT     =    0   MAX ØNLINE JØB RSRCE  MCØ    =   92
MAX ØNLINE JØB RSRCE  M9T     =    3   MAX ØNLINE JØB RSRCE  M7T    =    1
MAX ØNLINE JØB RSRCE  MSP     =    2   MAX ØNLINE JØB RSRCE  MBT    =    3
DFT ØNLINE JØB RSRCE  DCØ     =   32   DFT ØNLINE JØB RSRCE  D9T    =    0
DFT ØNLINE JØB RSRCE  D7T     =    0   DFT ØNLINE JØB RSRCE  DSP    =    0
DFT ØNLINE JØB RSRCE  DBT     =    0   TØTAL *BATCH*  RSRCE  TCØ    =  800
TØTAL *BATCH*  RSRCE  T9T     =    4   TØTAL *BATCH*  RSRCE  T7T    =    1
TØTAL *BATCH*  RSRCE  TSP     =    3   TØTAL *BATCH*  RSRCE  TBT    =    4
CURNT *BATCH*  RSRCE  CCØ     =    0   CURNT *BATCH*  RSRCE  C9T    =    0
CURNT *BATCH*  RSRCE  C7T     =    0   CURNT *BATCH*  RSRCE  CSP    =    0
CURNT *BATCH*  RSRCE  CBT     =    0   MAX BATCH  JØB RSRCE  MCØ    =   64
MAX BATCH  JØB RSRCE  M9T     =    4   MAX BATCH  JØB RSRCE  M7T    =    1
MAX BATCH  JØB RSRCE  MSP     =    2   MAX BATCH  JØB RSRCE  MBT    =    4
DFT BATCH  JØB RSRCE  DCØ     =   32   DFT BATCH  JØB RSRCE  D9T    =    0
DFT BATCH  JØB RSRCE  D7T     =    0   DFT BATCH  JØB RSRCE  DSP    =    0
DFT BATCH  JØB RSRCE  DBT     =    0   MAX GHØST  JØB SRVCE  MTIME  = 9999
MAX GHØST  JØB SRVCE  MLØ     = 9999   MAX GHØST  JØB SRVCE  MPØ    = 9999
MAX GHØST  JØB SRVCE  MDØ     = 9999   MAX GHØST  JØB SRVCE  MUØ    = 9999
MAX GHØST  JØB SRVCE  MPSTØ   = 1000   MAX GHØST  JØB SRVCE  MTSTØ  = 1000
MAX GHØST  JØB SRVCE  MFPØØ   =    6   MAX GHØST  JØB SRVCE  MTDIS  =32767
MAX GHØST  JØB SRVCE  MPDIS   =32767   DFT GHØST  JØB SRVCE  DTIME  = 9999
DFT GHØST  JØB SRVCE  DLØ     = 9999   DFT GHØST  JØB SRVCE  DPØ    = 9999
DFT GHØST  JØB SRVCE  DDØ     = 9999   DFT GHØST  JØB SRVCE  DUØ    = 9999
DFT GHØST  JØB SRVCE  DPSTØ   =   64   DFT GHØST  JØB SRVCE  DTSTØ  =   64
DFT GHØST  JØB SRVCE  DFPØØ   =    6   DFT GHØST  JØB SRVCE  DTDIS  =32767
DFT GHØST  JØB SRVCE  DPDIS   =32767   MAX ØNLINE JØB SRVCE  MTIME  = 9999
MAX ØNLINE JØB SRVCE  MLØ     = 9999   MAX ØNLINE JØB SRVCE  MPØ    = 9999
MAX ØNLINE JØB SRVCE  MDØ     = 9999   MAX ØNLINE JØB SRVCE  MUØ    = 9999
MAX ØNLINE JØB SRVCE  MPSTØ   = 1000   MAX ØNLINE JØB SRVCE  MTSTØ  = 1000
MAX ØNLINE JØB SRVCE  MFPØØ   =   18   MAX ØNLINE JØB SRVCE  MTDIS  =32767
MAX ØNLINE JØB SRVCE  MPDIS   =32767   DFT ØNLINE JØB SRVCE  DTIME  = 9999
DFT ØNLINE JØB SRVCE  DLØ     = 9999   DFT ØNLINE JØB SRVCE  DPØ    = 9999
DFT ØNLINE JØB SRVCE  DDØ     = 9999   DFT ØNLINE JØB SRVCE  DUØ    = 9999
DFT ØNLINE JØB SRVCE  DPSTØ   =   64   DFT ØNLINE JØB SRVCE  DTSTØ  =   64
```

Figure 9. Example of a CONTROL! Display

```
DFT ONLINE JOB SRVCE  DFP00 =     4    DFT ONLINE JOB SRVCE  DTDIS =32767
DFT ONLINE JOB SRVCE  DPDIS =32767    MAX BATCH  JOB SRVCE  MTIME =   999
MAX BATCH  JOB SRVCE  ML0   =  9999    MAX BATCH  JOB SRVCE  MP0   =  9999
MAX BATCH  JOB SRVCE  MD0   =  9999    MAX BATCH  JOB SRVCE  MU0   =  9999
MAX BATCH  JOB SRVCE  MPST0 =  1000    MAX BATCH  JOB SRVCE  MTST0 =  1000
MAX BATCH  JOB SRVCE  MFP00 =    18    MAX BATCH  JOB SRVCE  MTDIS =32767
MAX BATCH  JOB SRVCE  MPDIS =32767    DFT BATCH  JOB SRVCE  DTIME =     5
DFT BATCH  JOB SRVCE  DL0   =  9999    DFT BATCH  JOB SRVCE  DP0   =   999
DFT BATCH  JOB SRVCE  DD0   =  9999    DFT BATCH  JOB SRVCE  DU0   =  9999
DFT BATCH  JOB SRVCE  DPST0 =    64    DFT BATCH  JOB SRVCE  DTST0 =    64
DFT BATCH  JOB SRVCE  DFP00 =     4    DFT BATCH  JOB SRVCE  DTDIS =32767
DFT BATCH  JOB SRVCE  DPDIS =32767    SYM/FAUTH DFT GHOST   DCR   =     1
SYM/FAUTH DFT GHOST   DLP   =     1    SYM/FAUTH DFT GHOST   DCP   =     1
SYM/FAUTH DFT GHOST   DEQ   =     1    SYM/FAUTH DFT GHOST   DXX   =     1
SYM/FAUTH DFT GHOST   DRP   =     1    SYM/FAUTH DFT ONLINE  DCR   =     1
SYM/FAUTH DFT ONLINE  DLP   =     1    SYM/FAUTH DFT ONLINE  DCP   =     1
SYM/FAUTH DFT ONLINE  DEQ   =     1    SYM/FAUTH DFT ONLINE  DXX   =     1
SYM/FAUTH DFT ONLINE  DRP   =     1    SYM/FAUTH DFT BATCH   DCR   =     1
SYM/FAUTH DFT BATCH   DLP   =     1    SYM/FAUTH DFT BATCH   DCP   =     1
SYM/FAUTH DFT BATCH   DEQ   =     1    SYM/FAUTH DFT BATCH   DXX   =     1
SYM/FAUTH DFT BATCH   DRP   =     1
```

Figure 9. Example of a CONTROL! Display (cont.)

PARTITION DISPLAY AND DEFINITION COMMANDS

The CP-V multi-batch partition system was designed to maximize utilization of the system's resources. Job throughput is dependent upon the efficiency with which system resources (i.e., core, tape drives, disk pack spindles, etc.) are utilized. A crude measure of efficiency is the percentage of time that each device and the CPU are busy for a given work load over a given period of time. Efficiency goes up and throughput increases when the resource utilization is greater for a particular work load and time sample. For a varying work load, however, high throughput will not always result from simultaneously high usage of all system resources.

Greater efficiency may be realized by overlapping I/O functions. One method of accomplishing this is to allow several jobs to reside in core concurrently, each receiving a time slice. If the currently executing job issues an I/O call that causes physical I/O to occur, its quantum is ended and another job is scheduled and begins execution (i.e., it receives the usage of the CPU resource). Thus, two system resources, the I/O device called by the first job and the CPU, are now being utilized concurrently. It is easy to extrapolate from here to visualize several tape drives, a RAD, a disk pack, two line printers, and a card reader all operating simultaneously. The cost of achieving this overlap is, of course, more core since it is required for all processors whether I/O-bound or CPU-bound.

However, resource overlap will not occur if, say, three compute-bound jobs are scheduled for execution concurrently. Each job will, in turn monopolize the only resource all three need (the CPU) while other resources stand idle. This is why the Multi-Batch Scheduler (MBS) is needed. One of the main functions of MBS is to schedule jobs for concurrent execution so that they utilize as many resources as possible, and not to schedule jobs that will vie for a single resource, which would cause one or the other to occupy available core space (itself a resource) while waiting for a resource to be freed.[†] Ideally, a multi-batch scheduler would schedule a compute-bound job with several I/O bound jobs and would let the compute-bound job take up the CPU slack while the others wait for I/O to complete.

Thus far, the discussion of batch system performance has approached the subject of resource optimization based on only one criterion – gross work accomplished per time unit. If the total system work done over, say, a twenty-four hour period were the only consideration, the discussion might stop here. However, all installations have unique user requirements and operational procedures, and diverse machine configurations. Consequently, there are certain additional criteria on which system performance must be judged. These criteria might be termed operational considerations and with each of them is associated a priority that is higher than the one assigned to raw throughput.

A hypothetical illustration of an operational consideration might be an installation that has a system configuration utilizing six tape drives. Experience at this installation has shown that when a set of jobs that uses all drives comes up for execution, it is all the operator can do to mount and dismount the required tapes and respond to the messages that appear on the operator's console. Also, it is known that between the hours of 3:00 and 4:00 p.m. an inordinate number of small listing jobs are submitted for processing. Those jobs normally occupy the operator's entire time in separating the output. Therefore, the installation manager may wish to block execution of either the job set requiring six tape drives or the listing jobs between the hours of 3:00 and 4:00 p.m. A more common situation would be one in which an installation must guarantee fast turnaround on jobs

[†]MBS selects for execution the job with the highest priority and the longest time in the queue for which system resources are available.

of short duration that use minimal resources while jobs of long duration or those that use tape drives and private disk packs must be given a smaller share of the CPU time until the fast turnaround jobs are run. Both of these examples illustrate an important principle that emerges as a consequence of tempering raw throughput with operational considerations – submitted jobs must have attributes defined in terms of necessary maximum resources to run the job. This is necessary so the system may identify those attributes, categorize the job, and schedule it to be run so as to satisfy the operational considerations while guaranteeing maximum throughput.

Job attributes that are recognized by the Multi-Batch System are listed in Table 20. Those attributes that are used by the scheduler in determining when the job will run are indicated by a footnote.

It is the responsibility of the user to specify the attributes of his job on the LIMIT command so that his job will not be scheduled for execution in the same manner as one requiring a greater slice of the system's resources.

The system manager is able to allocate the resources of his system to jobs with certain attributes by defining batch partitions under which diverse categories of jobs may run. A partition is a collection of ranges of job attributes. In some systems, a partition is defined as a fixed, addressable area of core in which jobs with certain attributes may run. Partitions in CP-V are not that type. No physical system resources such as core, spindles, or tape drives are permanently allocated to a partition. All jobs executing in any partition draw their physical resource requirements from a common pool without regard to the partition under which they qualified for execution except that the numeric limits that pertain to that partition will apply. A list of attributes that comprise a partition may be found in Table 18.

When a partition does not have a job scheduled for it, the scheduler looks for qualified jobs for that partition and selects the one with the highest priority and the longest time in the queue of jobs waiting to be executed. Once in the system, the job may use any of the physical resources up to the limits assigned to its partition or (if the option was not specified) those specified on the LIMIT card.

In summary, partition definitions are a primary factor in the job selection process. The system manager may direct the power of his system to the categories of jobs he so chooses by means of those definitions.

Partition attributes are initially specified on the :PARTITION command of PASS2 in SYSGEN. A maximum of

Table 20. Job Resource Attributes

Attribute①	Description	Units
CORE②	Maximum core size required by job.	K words
TIME②	Maximum time that the job will require to execute.	minutes
9T②	Maximum number of 9-track drives that the job will require.	9T drives
7T②	Maximum number of 7-track drives that the job will require.	7T drives
SP②	Maximum number of disk pack spindles that the job will require.	spindles
Resource name③	Maximum amount of resource that the job will require.	-
ORDER②④	All previously entered jobs with this account must run prior to this job.	-
MOUNT②	Specifies which packs or tapes must be premounted. ⑤	numeric
ACCOUNT②	Specifies no other job with this account may run concurrently. ⑥	-

① Attributes are specified on the LIMIT command and the IPOOL and FPOOL commands.

② Attributes used by the MBS in selecting jobs for execution.

③ Resource name is specified on the :RES command at SYSGEN.

④ NORDER indicates job is not order-dependent.

⑤ Serial numbers are specified. Disk pack must be declared for shared or exclusive use.

⑥ Eliminates file-contention problems.

16 partitions may be defined for any system. It is recommended that 16 partition definitions be generated for all systems unless core memory is a serious consideration. This will provide a variety of job attribute classifications and those partitions in excess of the operational number may be locked from use (see LOCK, Table 18).

In a time-sharing/batch processing system, emphasis may be given to batch processing by opening up more partitions. However, it should be noted that CP-V is a queue-driven system and tasks are selected from prioritized queues without regard to the source of the request (i.e., on-line, batch, or remote batch). When there is a heavy on-line user load, as the number of batch partitions increases, the number of compute bound tasks increases and each receives a small fraction of the CPU time. This means that batch jobs will be able to get more CPU time because of larger quanta assigned to the batch partitions. This will not make a significant dent in on-line response time because interactive requests have a higher priority than compute bound jobs. More attention may be given to certain categories of batch jobs by increasing the number of partitions suitable for them. Note that when new partitions are opened up, BUM (Table 16) must be increased because it acts as a limit on the number of batch users. BUM may be greater than or equal to the number of open partitions without any adverse affects on performance, but if it drops below the number of open partitions, it will be the constraint on the number of jobs executing rather than the number of open partitions.

Partition definitions may be altered or displayed using Control. Since the definition tables are fixed-length once created by SYSGEN, additional partitions may not be created and existing partitions may not be deleted. However, existing partitions may be completely redefined or locked from further use. A description of the Control partition display and definition commands follows.

PARTITION This command passes execution control to the Partition sub-command level. It opens the partition definition tables for display if the user has a privilege level of at least 80, and for definition and display if the user has a privilege level of at least C0. This command must be given before any other partition-associated commands will be honored. Once given, no Control commands other than partition associated commands will be recognized until the BREAK key is depressed or an END command is issued. The Control partition definition stack (but not the partition definition tables) is cleared. This stack will retain any definitions subsequently specified until another PARTITION command is issued. The format for the command is

PA[RTITION]

Control then prompts for partition associated commands. An example is given in Figure 12, Example of Partition Commands.

DISPLAY n This command displays all of the attributes of partition n except for the account number of the job currently executing under the partition definition and the ID number of the current user. (Terminal page-width was the limiting factor. These parameters may be displayed by the "n attribute" command.) The format of the command is

$$\text{DI[SPLAY]}\left\{\begin{matrix}\text{n}\\ \text{n - n}\\ \text{ALL}\end{matrix}\right\}$$

where

n is the number of a valid batch partition (1-16).

n - n is a range of batch partition numbers.

ALL specifies that the attributes of all partitions are to be displayed.

In the example Figure 10, the attribute ranges of partition 1 are displayed. Any job requiring a tape or disk pack tape drive would not be able to run under this partition's definition. Jobs selected for this partition are to be held in core, if possible, rather than being swapped out.

```
>DISPLAY 1 (RET)

 PART   TIME   QUAN LOCK HOLD CUR  TOL  CO    7T    9T    SP
 ----  ------  ---- ---- ---- ---  ---  ----  ----  ----  ----
   1    1-5    2000  NO   YES   1   10  0-15  0-0   0-0   0-0

>
```

Figure 10. Example of the DISPLAY Command

n attribute Single attributes for a specified partition may be displayed by using this command. The format of the command is

$$\left\{\begin{array}{l} n \\ n - n \\ ALL \end{array}\right\} \text{attribute}$$

where

n is the number of a valid batch partition (1-16).

n - n is a range of batch partition numbers.

ALL specifies that the attribute is to be displayed for all partitions.

attribute specifies the name of an attribute (see the Control Name column of Table 18). All characters of the name must be entered.

Example:

>12 QUAN

12 QUAN = 400

>

In the above example, the value QUAN for partition 12 is displayed.

Example:

>ALL 7T

1 7T = 0-0

2 7T = 0-1

3 7T = 0-2

4 7T = 1-2

>

For this example, the permissible range for 7-track tape drives requested by the user is displayed for partitions 1-4. (There are only four partitions in the particular system.)

n attribute = number The value of an attribute for a specified partition may be set by performing a sequence of two commands. The first, described here, is known as the set command. The second is the STORE command which is described below. The format of the set command is given below. Note that the only blank allowed is between the n and attribute specifications.

$$\left\{\begin{array}{l} n \\ n - n \\ ALL \end{array}\right\} \text{attribute} \quad \text{number}$$

where

n is the number of a valid batch partition (1-16).

n - n is a range of batch partition numbers.

ALL specifies that the attribute is to be set for all partitions.

attribute specifies the name of an attribute. (See the Control name column of Table 18.) All characters of the name must be entered.

number specifies the new value or range of values for the attribute and must lie within the permissible range of values for the attribute (see Table 18).

The set command causes the value indicated by "number" to be stored temporarily in the partition definition stack. The STORE command causes the values in the partition definition stack to replace their corresponding values in the partition definition tables. (This point will be discussed further after the STORE command is described.) Any other partition commands may be issued between the set and STORE commands, including BREAK and PROCEED. However, execution of the CLEAR, STORE, QUIT, and END commands causes the partition definition buffers to be cleared and all previously set values to be lost.

An example of the set command will be given in the example for the STORE command.

STORE This command causes the resultant values of all previous set commands (see "n attribute = number") to be entered into the partition definition tables from the partition definition stack and clears the partition definition stack. The format of the command is

ST[ORE]

Example:

>5 SP

5 SP = 0-2

>5 SP = 2-2

>5 SP

5 SP = 0-2

>STORE

VALUES STORED

>5 SP

5 SP = 2-2

>

In this example, the spindle range for partition 5 is displayed and a new spindle range is set. On subsequent display, however, the original spindle range remains unchanged because the new value has not passed from the partition definition stack to the partition definition tables.

The STORE command finalizes the previous set command by entering a range of 0-2 into the spindle attribute of the partition 5 table entries. The new value is then displayed

Due to the fact that MBS cannot access the partition definition tables while they are being updated by Control, all set command values are stored in a stack until the user wishes to enter them as a group. While the values are being stored in the partition definition stack, the partition definition tables remain unlocked. It is only when the user issues a STORE command that the partition definition tables are locked and real data transfer takes place. The STORE command thus minimizes the time that partition definition tables need to be locked to MBS.

Each time a STORE command is issued, the job queue is searched and those jobs that do not qualify for execution under redefined partitions are rescheduled under other partitions. Jobs that previously did not qualify under any partition are scheduled, if possible, for execution under a redefined partition. The least amount of rescheduling is required when all redefined partition attribute values are set and a single STORE command is issued.

CLEAR This command clears the partition definition buffer. Values obtained from set commands following the previous QUIT, END, STORE, or CLEAR commands are lost. The command format is

CL[EAR]

Example:

```
>6 9T=1-4
>6 9T
 6 9T 0-2
>CLEAR
>STORE
 VALUES STORED
>6 9T
 6 9T=0-2
```

Here, the range of 9-track tape drives for partition 6 is set at 1-4 in the partition definition stack and the current range is displayed from the partition definition tables. The partition definition stack is cleared and, therefore, no new attribute values are entered in the tables. Subsequently to show this, the original range of 9-track tape drives is displayed for partition 6.

BREAK A single BREAK character may be issued while executing partition commands in order to temporarily return user control to the control command level. At the control command level, only non-partition commands (i.e., LIST, CONTROL, etc.) will be recognized and other control parameters may be modified and displayed. The partition sub-command level may be re-entered by execution of the PROCEED command (which will be discussed shortly) and none of the buffer values resulting from previous set commands will be lost. The partition sub-command level may also be re-entered by executing a QUIT command followed by a PARTITION command. However, all of the buffer values resulting from previous set commands will be lost.

Figure 11 is an example that contains the BREAK and PROCEED commands. In this example, execution TIME limits for partition 10 are set in the partition definition buffer and the ID of the current user under parition 3 is displayed. At this point, a single BREAK character is issued returning the user to control command level. Control parameters with their display flags turned on are then listed and displayed and a PROCEED command is executed returning control to the partition command level. The TIME range from the previous set command affecting partition 10 is then entered and the resultant value displayed.

END If issued at the partition sub-command level, this command clears the partition definition stack, closes the partition definition tables to Control access, and returns execution control to the control command level. To display or modify the partition definition tables again, a new PARTITION command must be invoked. The format of the command is

END

An example of the END command is given in Figure 12.

```
>10 TIME=0-4
>3 USER#
 3 USER#=24
>

-LIST
 BUC
```

Figure 11. Example of BREAK and PROCEED Commands

```
OUC

GUC

-CONTROL

CURRENT # BATCH USERS        (BUC=)  14        CURRENT # ON-LINE USERS        (OUC)=  28

CURRENT # GHOST JOBS         (GUC)=   3

-PROCEED

>STORE

 VALUES STORED

>10 TIME

 10 TIME=0-4

>
```

Figure 11. Example of BREAK and PROCEED Commands (cont.)

```
!CONTROL

CONTROL HERE

-PARTITION

>DISPLAY ALL

 PART   TIME      QUAN LOCK HOLD CUR  TOL   CO       7T    9T    SP
 ---- --------- ---- ---- ---- --- ---- ------- ----- ----- -----

   1  nnnn-nnnn† nnnn YES  YES  nnn nnnn nnn-nnn nn-nn nn-nn nn-nn
   2  nnnn-nnnn nnnn  NO   NO   nnn nnnn nnn-nnn nn-nn nn-nn nn-nn
   :
  16  nnnn-nnnn nnnn  NO   NO   nnn nnnn nnn-nnn nn-nn nn-nn nn-nn

>DISPLAY 10

 PART   TIME      QUAN LOCK HOLD CUR  TOL   CO       7T    9T    SP
 ---- --------- ---- ---- ---- --- ---- ------- ----- ----- -----

  10    2-15      500  NO   NO   19   83  16-24    0-1   0-2   0-2

>4 CO

 4 CO=4-12

>3 HOLD

 3 HOLD= 1
```

† The number of n's indicates the maximum number of digits that would print.

Figure 12. Example of Partition Commands

```
≥6 QUAN(RET)

 6 QUAN= 600

≥1 LOCK=1(RET)

≥2 LOCK=NO(RET)

≥8 SP=2-4

≥STORE(RET)

 VALUES STORED

≥8 SP(RET)

 8 SP= 2-4

≥END(RET)

-END(RET)

!
```

Figure 12. Example of Partition Commands (cont.)

MISCELLANEOUS COMMANDS

QUIT This command may be given after a BREAK command to terminate the interrupted session. If issued, the interrupted environment will be lost and execution will continue at the control command level. The format of the command is

QU[IT]

PROCEED This command returns control to the partition sub-command level following an interruption. If no interruption has occurred (BREAK key has not been depressed) and execution is at the control command level, the message "NO BREAK ISSUED" will be output and a prompt for a new command will be issued. The format of the command is

PR[OCEED]

An example of the PROCEED command is given in Figure 11.

BREAK At the control command level, depressing the BREAK key may serve to interrupt a lengthy display. After the first BREAK is issued, subsequent BREAKs are ignored by Control. Four successive BREAKs return control to TEL.

END At the control command level, this command causes an exit from the Control processor and returns control to TEL. The format of the command is

EN[D]

An example of this command is given in Figure 12.

TIME This command displays the time of day in hours and minutes. The format of the command is

TI[ME]

Example:

```
-TIME(RET)

 12:32
```

SAMPLE COMMAND SEQUENCE

The command sequence in Figure 13 shows how the Control program may be used to modify control values and to display and modify partition attributes.

CONTROL ERROR MESSAGES

The Control processor will return an error message to the terminal whenever a command is entered incorrectly. The error messages are listed in Table 21.

CONTROL COMMAND SUMMARY

Table 22 contains a summary of the control level commands for the Control processor. Table 23 contains a summary of the partition subcommand level commands for the Control processor. In both tables, the left-hand column specifies the format and the right-hand column defines the function.

!CONTROL RET	(calls CONTROL program)
CONTROL HERE	
-ADD RET	(requests to set item display flags)
.TB RET	(sets TB display flag)
.UB RET	(sets UB display flag)
. RET	(exits from ADD routine)
-CO RET	(displays items with flags on)
# CHARS TERMINAL BLOCK (TB)= 40 # CHARS TERMINAL UNBLOCK (UB)= 10	
-TB = 60 RET	(sets value of TB to 60)
-PART RET	(enters partition command level)
>DI ALL RET	(displays all partition attributes)

PART	TIME	QUAN	LOCK	HOLD	CUR	TOL	CO	7T	9T	SP
----	---------	----	----	----	---	----	------	-----	-----	------
1	0-5	400	NO	YES	25	211	0-15	0-0	0-0	0-0
2	0-15	450	NO	NO	13	77	0-32	0-0	0-2	0-1
3	15-60	500	NO	NO	7	14	16-63	0-0	0-3	0-2
4	6-15	600	YES	YES	0	3	16-32	0-0	0-0	0-0

>	(interrupts partition display session)
←	
-TI RET	(calls for time to be displayed)
23:12	
-PR RET	(proceeds at partition command level)
>4 LOCK = 0 RET	(sets LOCK attribute of partition 4 to 0 = No)
>STORE RET	(in effect, opens partition 4 for scheduling)
VALUES STORED	(stores attribute in partition table)
>END RET	(exits from partition command level)
-END RET	(exits from CONTROL)
!	

Figure 13. Sample Command Sequence

Table 21. Control Error Messages

Message	Description
'CURRENT' VALUES MAY NOT BE ALTERED	An attempt was made to set a current value (see Table 17). Current values may only be displayed.
ILLEGAL COMMAND	The command verb was not recognized by Control. Reenter the command.
INVALID NUMBER	The value of the control parameter is invalid. Reenter the control parameter name and the correct value.
I/O ERROR AT xxxxx THROUGH M:XX DCB 'message from ERRMSG file corresponding to ABN code' I/O SUBROUTINE CALL AT xxxxx	An ERR/ABN return occurred through reading the M:SI DCB or writing the M:LO or M:DO DCBs. Input and output will be reassigned to the default devices for this type of job if set to other devices. Otherwise an abort will occur.
LIMIT CHANGE REQUIRES AT LEAST PRIVILEGE B0	The user does not have privilege level of at least X'B0' which is required to modify system limits.
MONITOR ACCESS REQUIRES AT LEAST PRIVILEGE 80	The user does not have a privilege level of at least X'80' which is required to access the monitor.
NO BREAK ISSUED	An attempt was made to QUIT or PROCEED when no BREAK was active.
NO SUCH ATTRIBUTE	The partition attribute specified in the command does not exist.
NO SUCH CONTROL NAME	An attempt was made to change a performance parameter that does not exist or is incorrectly represented.
NO SUCH NAME	An attempt was made to set or reset a nonexistent print flag with an ADD or DROP command. Reenter the command with the correct flag.
NO SUCH PARTITION	The partition number specified in the command is out of the range 1-16 or refers to a nonexistent partition.
NUMBER NOT WITHIN LEGAL RANGE PERMISSIBLE RANGE = xx-xx	The value of the control parameter is outside of the legal range. Reenter the command with a legal control parameter value (see Table 16).
PARTITION ATTRIBUTE DEFINITION REQUIRES AT LEAST PRIVILEGE CO.	The user does not have a privilege level of at least X'CO' which is required to change partition attribute definitions.
PARTITION CONTROL ACTIVE	An attempt was made to reopen partition control without terminating the previous command session.
PARTITION CONTROL ACTIVE QUIT, PROCEED, OR CONTROL COMMAND	The user has attempted to reenter the partition subcommand level after having interrupted a previous partition command level session with a break. He must resume the previous session, quit it, or issue a control level command.
SAD CAL FAILURE IN '34MAP' – REASSEMBLE	An internal assembly problem exists. See system analyst.

Table 21. Control Error Messages (cont.)

Message	Description
****	A numeric overflow occurred on CONTROL or CONTROL! display of a system control parameter. The parameter will automatically be displayed as a single parameter display.

†SMUIS is the maximum number of users allowed in the system. (It is the sum of the SYSGEN parameters MAXB, MAXO, and MAXG.)

††OUM is the maximum number of on-line users allowed in the system.

†††MAXG is the maximum number of ghost jobs that may be present in the system and is specified at SYSGEN time.

Table 22. Control Command Summary (Control Level)

Command	Description
-AD[D] .item .item ⋮	Turns on the print flags for the specified items. An item may be the Control Name of any item in Tables 16 and 17.
BREAK (i.e.)	Returns control to TEL if depressed four consecutive times. (May be used to interrupt a lengthy display by issuing once.)
CO[NTROL]	Displays all parameters listed in Tables 16 and 17 that have their print flags on.
CO[NTROL]!	Displays all parameters listed in Tables 16 and 17 whether or not their print flags are on.
-DR[OP] .item .item ⋮	Turns off the print flags for the specified items. An item may be the Control Name of any item in Tables 16 and 17.
EN[D]	Causes an exit from the Control processor and returns control to TEL.
LI[ST]	Lists the names of all items which have their print flags on.
name	Displays the specified parameter value. The name must be an item name from the lists in Tables 16 and 17.
name = value	Sets the value of the specified control parameter. The name must be the Control Name of a control parameter and the value must be within the range for the parameter (see Table 16).
PR[OCEED]	Returns control to the partition sub-command level following an interruption. If no interruption has occurred (BREAK key has not been depressed), an error message is output and execution returns to the control command level.

Table 22. Control Command Summary (Control Level) (cont.)

Command	Description
QU[IT]	Terminates a previous session interrupted by a BREAK command and passes execution to the control level.
TI[ME]	Displays the time of day in hours and minutes.

Table 23. Control Command Summary (Partition Sub-Command Level)

Command	Description
-AD[D] .item .item ⋮	Turns on the print flags for the specified items. An item may be the Control Name of any item in Tables 16 and 17.
BREAK (i.e. [illegible])	Causes execution control to return to the control command level and to remain there until a PROCEED command is invoked or until a QUIT command followed by a PARTITION command is issued.
CL[EAR]	Clears the partition definition buffer.
$\text{DI[SPLAY]}\begin{Bmatrix}\text{n}\\ \text{n - n}\\ \text{ALL}\end{Bmatrix}$	Displays the partition attributes for the partition numbered "n", a range of partitions, or for all partitions. The partition attributes ACCT and USER# are not displayed.
-DR[OP] .item .item ⋮	Turns off the print flags for the specified items. An item may be the Control Name of any item in Tables 16 and 17.
EN[D]	Clears the partition definition buffer, closes the partition definition tables to Control access, and returns execution control to the control command level.
$\begin{Bmatrix}\text{n}\\ \text{n - n}\\ \text{ALL}\end{Bmatrix}\text{attribute}$	Displays the value of the specified attribute for partition number "n", a range of partitions, or for all partitions. "Attribute" must be the name of an attribute listed in Table 18.
$\begin{Bmatrix}\text{n}\\ \text{n - n}\\ \text{ALL}\end{Bmatrix}\text{attribute = number}$	Sets the value of the specified attribute for partition number "n", a range of partitions, or for all partitions. "Attribute" must be the name of an attribute listed in Table 18; "n" must be a valid partition number; "value" must be within the range for the attribute and must not cause the currently allocated resources for that attribute to exceed the maximum resource value.
PA[RTITION]	Clears the partition definition buffer and opens the partition definition tables for display (if the user privilege is at least X'80') or modification (if the user privilege is at least X'C0').
ST[ORE]	Enters the values of all attributes set in the partition definition buffer into the partition definition tables and clears the partition definition buffer.

STATS

The CP-V performance monitor, STATS, performs two functions:

- It displays selected performance data in real-time.
- It creates chronological and sorted "snapshot" records of performance data for later processing by the Summary processor.

A good procedure to follow for monitoring system performance is outlined below:

1. Have operations personnel start STATS as a ghost job each day to generate the snapshot files.
2. Use Summary to periodically process the snapshot files, obtaining chronological data, sorted and ordered data, and an overall summary of the system's performance.

The following types of data are typically extracted from the Summary reports:

1. A histogram of estimated user intensity showing the frequency of occurrence of each user intensity.
2. The percent of CPU time versus the estimated user intensity.
3. On-line processor usage.
4. Batch processor usage.
5. Monitor service versus the number of users logged for a given user intensity group.

STATS may be run as a batch job, as a ghost job, or on-line. The must must have a privilege level of at least 80. The DCBs used by STATS are listed in Table 24.

In the batch mode, STATS is called with the STATS control command. The series of cards that follow list the commands to STATS. When writing this list of commands the user must anticipate the interactive statements that STATS would send were it run in the on-line mode. This will be clear when the commands are described.

STATS is initiated as a ghost job by the following key-in:

    GJOB STATS

The commands to STATS must have been stored previously in a file named GHOSTSI in the :SYS account. A default GHOSTSI file is provided in :SYS. The output files GHOSTDO and GHOSTLO are created to contain the output lines normally output via M:DO and M:LO respectively. The files GHOSTDO and GHOSTLO are generated automatically in the :STATS account and the associated DCBs may not be SET or ASSIGNed. Symbiont output is not allowed for a STATS ghost job since the job may sleep for many hours.

The STATS processor is called on-line by entering STATS as a TEL command. The processor responds by typing 'STATS version number HERE' and then prompts for a command.

Example:

```
!STATS
STATS B00 HERE
-
```

Table 24. STATS DCB Usage

DCB	Function	On-line Default	Batch Default	Ghost Setting
M:SI ① ②	Command input and user responses to interactive statements.	User console	After !STATS card	fid=GHOSTSI. :SYS
M:LO ② ③	Report output.	User console	Line printer	fid=GHOSTLO. :STATS
M:DO ②	Interactive statements and error messages.	User console	Line printer	fid=GHOSTDO. :STATS

Notes:

① M:SI may be assigned to a file containing a predefined set of commands. This allows automatic execution of repetitious tasks. Input from a file will be echoed via M:DO to provide a hard copy.

② M:SI, M:LO, and M:DO may be assigned to separate files via on-line SET or batch ASSIGN commands except when STATS is run as a ghost job.

③ M:LO may be assigned to the line printer if the on-line user's account authorization contains the appropriate flags.

Table 24. STATS DCB Usage (cont.)

DCB	Function	On-line Default	Batch Default	Ghost Setting
F:2	Chronological snapshot file (consecutive).	fid= SNAPSHOT	fid= SNAPSHOT	fid= SNAPSHOT. :STATS
F:3	Sorted snapshot file (keyed).	fid= SSNAPSHOT	fid= SSNAPSHOT	fid= SSNAPSHOT. :STATS
M:OC	Ghost job error messages for operator.	None	None	Operator's Console

STATS prompts for commands with a dash (-) and prompts for responses to interactive statements with a 'greater than' (>) character.

Depressing the BREAK key once will interrupt any current operation and leave control at the STATS command level. The PROCEED command (described later) is used to resume execution from the point of interrupt.

KEY CONCEPTS

TERMINOLOGY

A number of definitions are necessary for an understanding of the STATS processor.

SNAPSHOT file a consecutive file with one record for each snapshot interval. The records are added chronologically. Each record contains an entry for each computed statistic (there are 100 such items), the SYSTEM and SWAP histograms (defined later), and a list of the processor names for which CPU time was computed individually.

SSNAPSHOT file a binary file that is identical to the data in the SNAPSHOT file except that the records are reordered. The reordering is done by using a key that is generated based upon the snapshot user intensity, number of logged on users, and the time and date.

User intensity an estimate of the on-line user load computed by dividing on-line CPU time by user think-type time. This number grows larger if the tasks are longer and the users do not take long to think and type.

Sample interval either one of the following depending upon the operation being performed:

1. The time between snapshots.

2. The time between displays.

ALL A term that is used to specify the interval from the time of system reset or start-up to the time of a report. ALL statistics are not as informative as other statistics since the entire interval is not always of interest. For example, the time between system start-up and the time the first user logs on is a period which may not be of interest.

TERMINAL INTERACTION CONCEPTS

There are several concepts that define the measurements associated with on-line user characteristics and system response. These measurements are depicted in Figure 14.

STATS REPORT FORMAT CONTROL

When the user requests reports or displays, STATS refers to a set of print flags to determine which statistical groups are to be printed. The print flags and other STATS flags are listed in Table 25. Several of the flags have been described in greater detail in Figures 15 through 26. (These figures are referenced throughout Table 25.) Two of the flags are not used in determining which statistical groups are to be printed; rather, they determine whether or not reports and snapshot files can be generated simultaneously. Commands are provided that enable the user to set and reset the flags.

The CONTROL! command is the only display command that does not reference print flags. Instead, it displays a subset of the items listed in Tables 16 and 17. The display is the same as the PARAM display group (see Figure 15) but is not controlled by the PARAM flag.

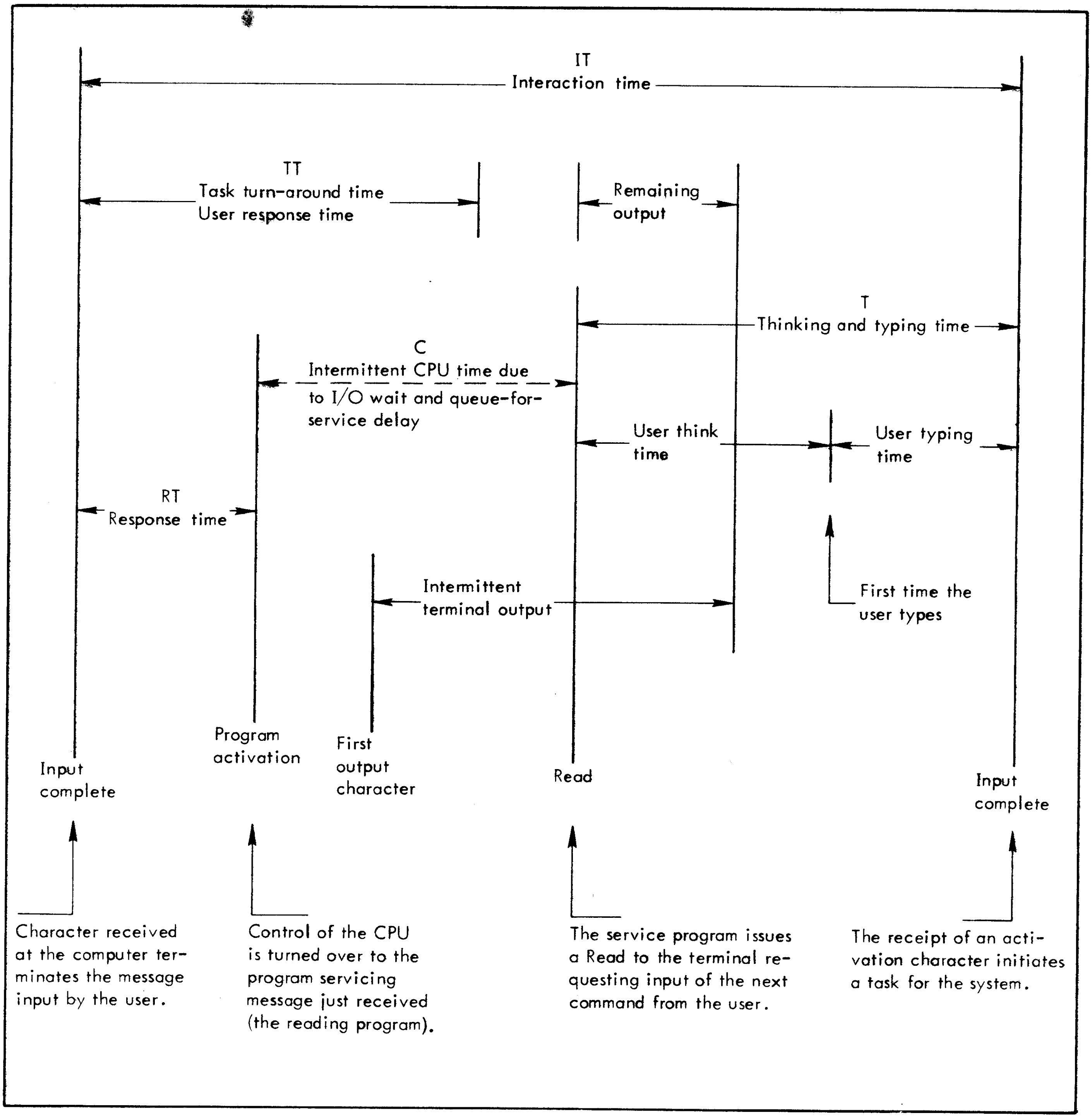


Figure 14. Terminal Interaction Concepts

Table 25. STATS Flags

Item name	Description
1	A standard display that lists histogram data as for the SYSTEM display group including response time, interaction time, turnaround time, task time, character in and character out histograms.
2	A standard display that includes the same information as standard display '1' and, optionally, histograms for a preselected shared processor. The shared processor may be selected by the system manager by setting P:PROCN via Executive Delta or a patch.

Table 25. STATS Flags (cont.)

Item name	Description
3	A standard display that includes the following display groups (which are described shortly): BATCH TASK (shown in order of appearance in the display) I/O USERS ONLINE QUEUE
4	A standard display that includes the SUMMARY and CPU display groups. This is the default standard display.
5	A standard display of the SWAP display group.
ALL	A display which includes standard displays 1, 3, 4, and 5 as follows: SUMMARY CPU QUEUE (shown in order of appearance in the display) ONLINE BATCH SYSTEM I/O TASK SWAP INTERACT USERS
BATCH	A display that lists the percent of CPU time used by each monitored shared processor for batch programs only. The CPU percentage includes batch execution and service time. (Figure 18 describes this display in detail.)
CPU	A display group that includes the percent of CPU time for the monitor and for on-line and batch users broken down into execution and service time. Monitor service, idle, and swap wait time is also listed for a complete view of CPU utilization. (Figure 17 describes this display in detail.)
C:CIT	A display of the total number of interactions received since start-up.
C:TIC	A display of the minutes since system start-up.
I/O	A display of the I/O rates for the system including symbiont accesses, CALs, and number of interactions per sample minute. (Figure 21 describes this display in detail.)
ONLINE	A display of the percent of CPU time used by on-line users for each of the monitored shared processors. The CPU percentage includes user execution and service time. (Figure 19 describes this display in detail.)
PARAM	A display that includes some control parameters and some other statistics. This print flag adds these control parameters to all subsequent reports. The CONTROL! command may be used to display this group. (Figure 15 describes this display in detail.)
PROC	A display of five histograms for a preselected processor including think-type time, turn-around time, CPU task time, terminal input characters per line, and terminal output characters per line. The processor is preselected by setting P:PROCN via Executive Delta or a patch. (Figure 25 describes this display in detail.)
QUEUE	A display that summarizes the user state queues at the end of each sample interval. (Figure 23 describes this display in detail.)
REPORT	A nondisplay flag that controls whether or not a report is to be generated while executing the FILE command.

Table 25. STATS Flags (cont.)

Item name	Description
SNAPSHOT	A nondisplay flag that controls whether or not snapshot records are to be generated while executing the DISPLAY command.
SUMMARY	A display that contains key measures of system performance (such as the number of logged users and the measures of response). (Figure 16 describes this display in detail.)
SWAP	A display of five histograms including distribution of: • The number of users per outswap. • The outswap time. • The inswap time for user JITs, shared processors, and overlap. • The inswap time for the entire user excluding the JIT and shared processors. • The inswap time for the entire user excluding shared processors and overlap. (Figure 26 describes this display in detail.)
SYSTEM	A display of seven histograms including distributions for response time, interaction time, think-type time, turn-around time, CPU task time, terminal input characters per line, and terminal output characters per line. (Figure 24 describes this display in detail.)
S:CUIS	A display of the number of current users in the system. This number includes batch users, on-line users, ghost jobs, and user's with lines who have not yet logged on.
TASK	A display of task statistics including interaction time, response time, and task time. (Figure 22 describes this display in detail.)
USERS	A display that lists the number of users for several shared processors. The numbers are the counts existing at the end of the sample interval. (Figure 20 describes this display in detail.)

The PARAM display is printed in the following format:

```
PARAM
HOUR:MINUTES                  INTERVAL IN MINUTES
MAX # BATCH USERS             MAX # ONLINE USERS
AVERAGE BATCH SIZE K          AVERAGE ONLINE SIZE K
# CHAR TERM BLOCK             # CHAR TERM UNBLOCK
MSEC W/O INTERRUPT            MSEC SWAP QUANTUM
AVE BATCH QUANTUM             MSEC ONLINE COMPUTE
COMPUTE QUEUE SHARING
```

where

HOURS:MINUTES specifies the hour followed (without a separator) by the minutes. This item is useful in selecting a sort filter for the Summary processor.

INTERVAL IN MINUTES is the number of minutes in the snapshot interval. This item is useful in selecting a sort filter for the Summary processor.

Figure 15. Description of the PARAM Display

MAX # BATCH USERS   is the maximum number of concurrent batch users and is the control parameter BUM which may be set by the system manager via the Control processor.

MAX # ONLINE USERS   is the maximum number of concurrent on-line users and is the control parameter OUM which may be set by the system manager via the Control processor.

AVERAGE BATCH SIZE K   is the current average size of all active batch partitions. This number includes context and excludes shared processors and overlap. (Reference: UB:PCT table.)

AVERAGE ONLINE SIZE K   is the current average size of all on-line users. This number includes context and excludes shared processors and overlap. (Reference: UB:PCT table.)

#CHAR TERM BLOCK   is the number of characters at which to block terminal output. This is the control parameter TB which may be set by the system manager via the Control processor. This parameter should be set high enough to prevent terminal output-bound users from being swapped too frequently. The faster the terminal transmission, the higher this parameter should be set. If the value is too high, however, the allocated COC buffers may be insufficient to handle the load.

CHAR TERM UNBLOCK   is the number of characters at which terminal output is unblocked and is the control parameter UB which may be set by the system manager via the Control processor. This parameter causes output-bound users to be queued for CPU service before the COC output buffer for that user is empty. The faster the terminal, the higher this parameter should be set.

MSEC W/O INTERRUPT   is the amount of uninterrupted compute time guaranteed a user and is the current value of the control parameter QMIN which may be set by the system manager via Control. A low value increases the burst I/O rate. A high value decreases the rate at which the system services different users.

MSEC SWAP QUANTUM   is the amount of time a user is guaranteed core residency before swap out. This parameter is the current value of the control parameter SQUAN which may be set by the system manager via Control. Increasing this value decreases the swapping rate. This parameter can be used to reduce the swapper load, but it will be at the expense of user response time.

AVE BATCH QUANTUM   is the average setting of the QUAN control parameter for all active batch partitions. (QUAN is the time-slice by which compute-bound users are shared.) The QUAN parameter is set for each partition by the system manager via the Control processor.

MSEC ONLINE COMPUTE   is the setting of the QUAN control parameter for on-line users. (QUAN is the time-slice by which compute-bound users are shared.) The QUAN parameter is set by the system manager via the Control processor. The values of QUAN for on-line and batch and the number of on-line and batch users in the compute-bound queue determine the mix of on-line and batch computing that is done.

COMPUTE QUEUE SHARING   is the batch bias. Zero indicates that batch compute-bound tasks have less priority than on-line compute-bound tasks. Nonzero indicates that batch and on-line compute-bound tasks have equal priority. Batch bias is the BB control parameter that may be set by the system manager via the Control processor.

Figure 15. Description of the PARAM Display (cont.)

The SUMMARY display group is printed in the following format:

```
SUMMARY              ALL†    SNAP††
BATCH STREAMS
ONLINE USERS
% CPU/BATCH USER
% CPU/ONLINE USER
BATCH EXEC/SERV
ONLINE EXEC/SERV
CPU MSEC PER I/O
ONLINE TIME MIX
ONLINE INTENSITY
ONLINE TASKS/MIN
% INTERACTIVE
90% RESPONSE MSEC
TURNAROUND     SEC
ETMF
```

where

BATCH STREAMS is the number of active batch partitions at the end of the snapshot interval. This statistic is available for the SNAP column only.

ONLINE USERS is the number of on-line users at the end of the snapshot interval. This statistic is available for the SNAP column only.

% CPU/BATCH USER the percent of time used for batch execution and batch service divided by the number of batch users. This statistic is available for the SNAP column only.

% CPU/ONLINE USER the percent of time used for on-line execution and on-line service divided by the number of on-line users. This statistic is available for the SNAP column only.

BATCH EXEC/SERV is batch execution time divided by batch service time. This is a measure of the degree to which batch jobs are compute-bound.

ONLINE EXEC/SERV is on-line execution time divided by on-line service time. This is a measure of the degree to which on-line jobs are compute-bound.

CPU MSEC PER I/O is the sum of batch and on-line execution and service time (in milliseconds) divided by the number of SIOs. This is a measure of the degree to which batch and on-line jobs are compute-bound.

ONLINE TIME MIX is a measure of the compute-bound character of the on-line user load. It is obtained by dividing the compute-bound milliseconds by the interactive task milliseconds. This measure is large when the interactive load is light. A task is considered interactive when less than SQUAN milliseconds (see Table 16) are required for completion of a task.

†ALL lists statistics for the period of time since system restart.

††SNAP lists statistics for the sample interval only.

Figure 16. Description of the SUMMARY Display

ONLINE INTENSITY is an estimate of the on-line user load. It is computed by dividing the on-line CPU milliseconds by the number of think-type minutes. On-line CPU milliseconds include execution and service time for on-line users. This statistic is approximately equal to the average task time (in milliseconds) multiplied by the task rate (measured in tasks per think-type minutes).

ONLINE TASKS/MIN is a measure of the rate at which tasks are arriving from logged on-line users. It is computed by dividing the number of tasks received from on-line users by the total number of logged minutes for on-line users.

% INTERACTIVE is another measure of the character of the user load. It is computed by dividing the number of interactive tasks that completed within SQUAN milliseconds of CPU time by the total number of tasks that completed in the sample time interval and converting this value to a percent by multiplying by 100.

90% RESPONSE TIME defines the point on the response time histogram (the point representing a number of milliseconds) at which 90% of all tasks receive CPU service within this number of milliseconds.

AVERAGE TURNAROUND is the average number of seconds between the receipt of an activation character from a user and the first line of output for that user.

EXECUTION MULTIPLICATION FACTOR is a dynamically changing estimate of the execution time multiplication factor (ETMF). The user may multiply the amount of CPU time required for a task by the ETMF to estimate the elapsed time required to complete that task in that user environment.

Figure 16. Description of the SUMMARY Display (cont.)

The CPU display group is printed in the following format:

```
 CPU %          ALL†     SNAP††
BATCH EXEC
BATCH SERV
ONLINE EXEC
ONLINE SERV
MONITOR SERV
IDLE
SWAP WAIT
I/O WAIT
I/O&SWP WAIT
TOTAL
```

where

BATCH EXEC is the percent of CPU time spent for batch programs executing in the mapped slave mode.

BATCH SERV is the percent of CPU time spent for the monitor for services required by batch programs. User service for batch programs is always executed in the mapped master mode.

†ALL lists statistics for the period of time since system restart.

††SNAP lists statistics for the sample interval only.

Figure 17. Description of the CPU Display

ON-LINE EXEC is the percent of CPU time spent for on-line users executing in the mapped slave mode.

ON-LINE SERV is the percent of CPU time spent in the monitor for services required by on-line users. User service for on-line users is always executed in the mapped master mode.

MONITOR SERV is the percent of CPU time spent for the monitor for scheduling, swapping, symbionts, and other monitor services executing in the unmapped master mode.

IDLE is the percent of CPU time spent for unmapped master mode with no execution and no swaps in progress.

SWAP WAIT is the percent of CPU time spent in unmapped master mode with a swap in progress but with no execution in progress.

I/O WAIT is the percent of CPU time spent in unmapped master mode with I/O in progress but with no execution and no swapping in progress.

I/O & SWP WAIT is the percent of CPU time spent in unmapped master mode with a swap and I/O in progress but with no execution in progress.

TOTAL is the sum of the percentages listed above. The accounting for ghost jobs causes this value to be less than 100%. Deferred accounting may cause a value of less than 100% to be followed by a value greater than 100% on the next interval.

Figure 17. Description of the CPU Display (cont.)

The BATCH display group prints in the following format:

```
 BATCH %    ALL[t]    SNAP[tt]
LINK
DELTA
:P00
:P11
EDIT
PCL
BASIC
METASYM
LOADER
FORT
USER
SHARED
```

where the percent of the CPU time for batch users is listed for each of the monitored shared processors. All programs which are not shared are grouped and listed as USER PROG. All other shared processors are grouped and listed as OTHER SHARED.

[t] ALL lists statistics for the period of time since system restart.

[tt] SNAP lists statistics for the sample interval only.

Figure 18. Description of the BATCH Display

The ON-LINE display group prints in the following format:

```
 ONLINE %     ALL†     SNAP††
LINK
DELTA
:P00
:P11
EDIT
PCL
BASIC
METASYM
LOADER
FORT
USER
SHARED
```

where each of the monitored shared processors (as defined by SYSGEN) is named and the percent of the CPU time used by each processor is listed. The percent is based upon the total available CPU time in the interval. All programs that are not shared are listed as USER PROG. All other shared processors are grouped and listed as OTHER SHARED.

†ALL lists statistics for the period of time since system restart.

††SNAP lists statistics for the sample interval only.

Figure 19. Description of the ON-LINE Display

The USERS display group prints in the following format:

```
 USERS     #
BASIC
DELTA
EDIT
FDP
FORTRAN
FORTLIB
LINK
LOADER
METASYM
PCL
TEXT
IN CORE
```

where each of the listed processors are associated with the number of users printed at the right. The number of users is evaluated only at the end of each snapshot interval and does not represent an average count for the interval. The number of users in core (item IN CORE) is a count at the end of the interval and includes STATS as one of the users in core. All these statistics in this group are counts (not averages) that provide some understanding of the system performance.

Figure 20. Description of the USERS Display

The I/O display group prints in the following format:

```
 I/O PER MIN    ALL†     SNAP††
SERVICE REQ
INTERACTIONS
CHAR IN
CHAR OUT
TERM WRITES
I/O ACCESSES
SYMBIONT
IN SWAPS
OUT SWAPS
```

where

SERVICE REQ is the average number of system CALs executed per minute of sample time.

INTERACTIONS is the average number of tasks submitted by on-line users per minute of sample time. Each task is initiated by an activation character such as a carriage return.

CHAR IN is the average number of characters received by the COC per minute of sample time. The average number of characters per line may be obtained by dividing the CHAR IN value by the INTERACTIONS value.

CHAR OUT is the average number of characters output by the COC per minute of sample time.

TERM WRITES is the number of terminal writes, i.e., the number of lines output to the terminal. The average number of characters per terminal write may be computed by dividing the CHAR OUT value by the TERM WRITES value.

I/O ACCESSES is the average number of tape, RAD, and disk accesses per minute of sample time. This rate may be limited by the number of core partitions, by the capacity of an I/O device or channel, or by low user demand.

SYMBIONT is the average number of symbiont accesses per sample minute. This rate tends to be device controlled and buffered from user demands.

IN SWAPS is the average number of users swapped in per minute of sample time.

OUT SWAPS is the average number of outswaps per minute of sample time. More than one user may be swapped out in a single out swap.

†ALL lists statistics for the period of time since system restart.

††SNAP lists statistics for the sample interval only.

Figure 21. Description of the I/O Display

The TASK display group has the following format:

```
TASK        ALL†    SNAP††
INTERACT
THINK-TY
TURNARND
COMPLETE
RESPON MS
CPU MS
```

where

INTERACT is the average number of seconds between activation characters.

THINK-TY is the average number of seconds between the prompt character transmission and the receipt of activation character.

TURNARND is the average number of seconds from the receipt of the activation character to the first line of output printed to terminal.

COMPLETE is the average number of seconds from the receipt of the activation character to the next prompt character transmission.

RESPON is the average number of milliseconds from the receipt of the activation character to the beginning of the first quantum of CPU time.

CPU is the average amount of CPU time required for all tasks completed in the sample interval.

†ALL lists statistics for the period of time since system restart.

††SNAP lists statistics for the sample interval only.

Figure 22. Description of the TASK Display

The QUEUE group display has the following format:

```
QUEUE     #
LOGGED
LOGGING
BATCH
GHOST
TERM IN
TERM OUT
COMPUTE
COMP BND
I/O
SLEEP
```

where

LOGGED is the number of on-line users who have already logged on; i.e., the number of on-line users who are not associated with the LOGON processor at the end of the sample.

LOGGING is the number of on-line users who have lines but are not yet logged on; i.e., the number of on-line users who are associated with the LOGON processor at the end of the sample.

Figure 23. Description of the QUEUE Display

BATCH is the number of active batch users at the end of the sample.

GHOST is the number of active ghost jobs at the end of the sample.

TERM IN is the number of terminal input bound users that are in the STI and STIO state queues. These users are thinking and typing.

TERM OUT is the number of terminal output bound users in the STOB and STOBO state queues. These users are waiting for output printing to be completed.

COMPUTE is the number of users in the SEC, SBK, SON, SIR, STOC, SIOC, SIOC2, SC, and SCU state queues. These users are waiting for CPU time but are not compute bound.

COMP BND is the number of users in the SCOM and SBAT state queues. These users are on-line and batch compute bound users.

I/O QUEUES is the number of users in the SIOW, SIOIP, SIOIP2, and SQA state queues.

SLEEPING is the number of users in the SW queue. These users have executed the M:WAIT system CAL.

Note: The total number of users for LOGGED, LOGGING, BATCH, and GHOST should equal the total number of users for TERM INPUT, TERM OUTPUT, COMPUTE, COMPUTE BND, I/O QUEUES, and SLEEPING. All of these queue statistics are counts at the end of the sample interval when STATS is in core and are not averages in any sense.

Figure 23. Description of the QUEUE Display (cont.)

The SYSTEM display includes seven system histograms in the following format:

```
                  <-------SYSTEM-------->                       <-SYSTEM->
                  RESP INTR TYPE TURN CPU                       INLN OTLN
Value of x         MS  SEC  SEC  SEC  MS       Value of x       CHAR CHAR
         <1                                            <5
         <2                                            <10
         <5                                            <15
        <10                                            <20
        <20                                            <25
        <50                                            <30
       <100                                            <35
       <200   (nn <-- the percent of response time     <40
       <500           entries that took less than      <45
        <1K           200 milliseconds but greater     <50
        <2K           than 100 milliseconds)           <55
        <5K                                            <60
       <10K                                            <65
        &UP                                            &UP
       TOT#                                            TOT#
```

where

RESP is a listing of the response time histogram. Each column entry specifies the percent of the on-line interactions that received CPU service within x milliseconds of the receipt of an activation character. The value of x is listed in the column at the left side. This distribution is dependent upon the swap device characteristics and upon the time required to find free core.

INTR is a listing of the interaction time histogram. Each column entry specifies the percent of the interactions that were received within x seconds of the last interaction for the user. The value of x is listed in

Figure 24. Description of the SYSTEM Display

the column at the left side. This distribution depends upon the rate at which users submit interactions and upon the rate at which the system completes the tasks.

TYPE is a listing of the think-type time histogram. Each column entry specifies the percent of interactions that required less than x seconds from time of the prompt character printing to the time at which the activation character is received. (The activation character is usually a carriage return.) The value of x is listed in the column at the left side. This histogram is dependent entirely upon the users thinking and typing time.

TURN is a listing of the turn-around time histogram. Each column entry specifies the percent of interactions that required less than x seconds from the receipt of an activation character to the time at which the user received the first line of output. The value of x is listed in the column at the left side. This histogram is dependent upon the response time and upon the account of time required to give the user some output. Some interactions generate output immediately while others generate output only upon completion of the task.

CPU is a listing of the CPU task time histogram. Each column entry specifies the percent of the interactions that required less than x milliseconds of CPU time. The value of x is listed in the column at the left side. Some tasks may require minutes. Other tasks may require only a few milliseconds. The distribution is dependent upon the type of tasks that users initiate.

INLN is a listing of the characters per terminal input line histogram. Each column entry specifies the percent of interactions which contained less than x characters. The value of x is listed in the column at the left side. This histogram is dependent upon the users' terminal usage. Type-ahead lines are concatenated as a single input without being counted individually.

OTLN is a listing of the characters per line of terminal output histogram. Each entry shows the percent of output lines that contained less than x characters. The value of x is listed in the column at the left side. The terminal block control parameter (TB) determines the number of characters at which the user will be suspended (become output bound).

Figure 24. Description of the SYSTEM Display (cont.)

The PROC display includes five histograms for a pre-selected shared processor which may be under test. The PROC display has the following format:

```
                 <--PROCESSOR-->                      <PROCESS->
                 TYPE  TURN  CPU                      INLN  OTLN
                  SEC   SEC   MS                      CHAR  CHAR
Value of x  <1                       Value of x  <5
            <2                                   <10
            <5                                   <15
           <10                                   <20
           <20                                   <25
           <50                                   <30
          <100                                   <35
          <200                                   <40
          <500                                   <45
           <1K                                   <50
           <2K                                   <55
           <5K                                   <60
          <10K                                   <65
           &UP                                   &UP
          TOT#                                  TOT#
```

The definitions of the columns are the same as for the SYSTEM display. Note that the CPU histogram and the RESP histogram are not included in the PROC display.

Figure 25. Description of the PROC Display

The SWAP display groups includes five histograms of swapper characteristics in the following format:

```
                    <--OUT SWAP--->                   <---IN SWAP--->
                                  TOT                 JIT REST   TOT
Value of x     #USERS             MS    Value of x    MS   MS    MS
        =0               <1               <1
        =1               <2               <2
        =2               <5               <5
        =3              <10              <10
        =4              <20              <20
        =5              <50              <50
        =6             <100             <100
        =7             <200             <200
        =8             <500             <500
        =9              <1K              <1K
       =10              <2K              <2K
       =11              <5K              <5K
       =12             <10K             <10K
       &UP              &UP              &UP
      TOT#             TOT#             TOT#
```

where

#USERS is a listing of the number of users per outswap histogram. Each column entry specifies the percent of the outswap attempts for which x users were swapped. The value of x is listed in the column at the left side. When x=0, no outswap occurred.

TOT is a listing of the outswap time histogram for actual outswaps. Each entry specifies the percent of outswaps that required less than x milliseconds. The value of x is listed in the column at the left side.

JIT is a listing of the time histograms that record the milliseconds required to swap in single page JITs, system overlays, and shared processors. Each column entry specifies the percent of the inswaps that required less than x milliseconds. The value of x is listed in the column at the left side. This histogram is dependent upon the swap device, and upon the size of the overlap and shared processors.

REST is a listing of the time histograms recording the milliseconds required to swap in the rest of a user once the JIT has been swapped in and processed. Each entry specifies the percent of user inswaps that required less than x milliseconds to be swapped in (exclusive of the JIT inswap). The value of x is listed in the column at the left side.

TOT is a listing of the time histograms recording the milliseconds required to swap in the entire users (JIT and REST above). Each column entry specifies the percent of inswaps that required less than x milliseconds. The value of x is listed in the column at the left side. The JIT and REST histograms may <u>not</u> be added together in any way to obtain this histogram.

Figure 26. Description of the SWAP Display

STATS COMMANDS

The STATS commands will be described in the following order:

HELP	CONTROL!
ADD	FILE
DROP	BREAK (i.e. ⓢ)
BUILD	PROCEED
LIST	TIME
DISPLAY	END

Descriptions of the commands and some of the examples are written for the on-line mode. If STATS is expecting a command and an asterisk is input in the first column, the remainder of the line is considered to be a comment.

HELP This command lists a brief description of STATS processor commands and DCB usage. The listing is shown in Figure 27. The format of the command is

HELP

```
  DESCRIPTION OF STATS COMMANDS
ADD          -  ADD PRINT FLAGS
B 0          -  RESET PRINT FLAGS
B(UILD) 1    -  BUILD DISPLAY FOR SYSTEM DISTRIBUTIONS
B 2          -  BUILD DISPLAY FOR SYSTEM AND PROCESSOR
                DISTRIBUTIONS
B 3          -  BUILD DISPLAY FOR 7 DISPLAY GROUPS
B 4          -  BUILD DISPLAY FOR SUMMARY (DEFAULT)
B 5          -  BUILD DISPLAY FOR SWAP DISTRIBUTIONS
CONTROL!     -  LIST SOME OF CONTROL PARAMS
DISPLAY      -  REPORT (AND UPDATE SNAPSHOT FILES)
DROP         -  DROP PRINT FLAGS
END          -  TERMINATE LISTS OR RETURN TO TEL
FILE         -  UPDATE SNAPSHOT FILES (AND REPORT)
LIST         -  LIST CURRENT FLAG STATUS
TIME         -  PRINTS CURRENT TIME
X            -  EXIT TO TEL
  DCB USAGE
M:SI - INTERACTIVE INPUT COMMANDS
M:LO - REPORTS
M:DO - INTERACTIVE OUTPUT
F:2  - CONSECUTIVE SNAPSHOT FILE
F:3  - KEYED SSNAPSHOT FILE
```

Figure 27. Format of the HELP Command Listing

ADD This command turns on the flags for the specified items. An item may be the name of any item (other than standard display numbers) in Table 25. When the STATS processor is entered, SUMMARY and CPU are on by default. The item names are entered following prompts. A null response (or END) to a prompt terminates the list of items. The format of the command is

ADD

Example:

```
-ADD (RET)
 ENTER ITEMS TO BE ADDED
>I/O (RET)
>CPU (RET)
>TASK (RET)
>END (RET)
-
```

The ADD command enables the user to select a group of parameters to be displayed whenever the DISPLAY command is given.

DROP This command turns off the flags for the specified items. An item may be the name of any item (other than standard display numbers) in Table 25. The item names are entered following prompts. A null response (or END) to a prompt terminates the list of items. The format of the command is

DROP

Example:

```
-DROP (RET)
 ENTER ITEMS TO BE DROPPED
>I/O (RET)
>CPU (RET)
>TASK (RET)
>END (RET)
-
```

BUILD This command turns on the print flag for a numbered standard display and turns off all other print flags. The format of the command is

B[UILD] n

where n is a decimal digit identifying the standard display for which the print flag is to be turned on. (The standard displays are listed in Table 25.) If n is the digit 0, all print flags are turned off. If more than one digit is specified, the last digit is used. The print flag for standard display '4' is ON when STATS is initially entered.

LIST This command lists the names of the STATS flags (other than standard display numbers) and their current status (on or off). It can be used to verify items to be included in a standard display or to make certain that a series of ADD and DROP commands had the desired effect. The format of the command is

LIST

Examples:

1. When the LIST command is given before any ADD or DROP commands have been given, the following default settings for STATS flags are listed.

```
-LIST (RET)
 FLAG      -   STATUS
PARAM      -      OFF
C:CIT      -      OFF
C:TIC      -      OFF
S:CUIS     -      OFF
SUMMARY    -  ON
CPU        -  ON
BATCH      -      OFF
ONLINE     -      OFF
USERS      -      OFF
I/O        -      OFF
TASK       -      OFF
QUEUE      -      OFF
SYSTEM     -      OFF
PROC       -      OFF
SWAP       -      OFF
SNAPSHOT   -      OFF
REPORT     -      OFF
```

2. In this example, the flag for standard display 3 is turned on and all other print flags are turned off. The LIST command is then used to obtain the current status of all print flags.

```
-BUILD 3 (RET)
-LIST (RET)
 FLAG     -   STATUS
PARAM     -      OFF
C:CIT     -      OFF
C:TIC     -      OFF
S:CUIS    -      OFF
SUMMARY   -      OFF
CPU       -      OFF
BATCH     -  ON
ONLINE    -  ON
USERS     -  ON
I/O       -  ON
TASK      -  ON
QUEUE     -  ON
SYSTEM    -      OFF
PROC      -      OFF
SWAP      -      OFF
SNAPSHOT  -      OFF
REPORT    -      OFF
```

DISPLAY This command displays (via M:LO) current statistics for those items (from the group of items listed in Table 25) that have their print flags on. The user specifies the number and frequency of displays by responses to two STATS interactive statements. The format of the command is

DISPLAY

Example:

```
-DISPLAY (RET)
 ENTER INTERVAL IN MINUTES
>5 (RET)
 ENTER # OF INTERVALS
>4 (RET)
```

If the output device cannot finish printing one display before another is ready, an error diagnostic is typed saying 'CANNOT MAINTAIN INTERVAL'. The next interval includes all time from the last printout until the next printout is able to begin.

Snapshot files may be generated during DISPLAY command execution if the SNAPSHOT flag is on.

CONTROL! This command displays all control parameters that are critical to system performance. (The group of parameters displayed is the same as for the PARAM display.) The format of the command is

C[ONTROL]!

The display immediately follows the command.

FILE This command creates snapshot records for the SNAPSHOT and SSNAPSHOT files. A report may be generated simultaneously by turning the REPORT flag and the desired print flags on. The format of the FILE command is

FILE

Example:

```
-FILE (RET)
 ENTER INTERVAL IN MINUTES
>1 (RET)
 ENTER # OF INTERVALS
>4
 SNAPSHOT FILES OPENED
..SLEEPING
 RECORD # 1, 16:29 SEP 19,'73
..SLEEPING
 RECORD # 2, 16:30 SEP 19,'73
..SLEEPING
 RECORD # 3, 16:31 SEP 19,'73
..SLEEPING
 RECORD # 4, 16:32 SEP 19,'73
..SLEEPING
 RECORD # 5, 16:33 SEP 19,'73
 SNAPSHOT FILES COMPLETED
```

BREAK This command interrupts any current STATS operation and leaves control at the STATS command level. The command consists of depressing the BREAK key once. The PROCEED command is used to resume execution from the point of interrupt. Print flags may be altered via the ADD and DROP commands before resuming execution via the PROCEED command.

PROCEED This command continues STATS interruption from a point at which the processor was interrupted by the BREAK key. The format of the command is

PROCEED

TIME This command displays the time and date via M:DO as received from the monitor via the M:TIME system CAL. The format of the command is

TIME

Example:

-TIME (RET)

15:10 OCT 02, '71

END This command causes an exit to TEL. The format of the command is

END

Example

-END (RET)

!

SAMPLE STATS SESSIONS

Examples 1 through 3, below, are sample STATS sessions that perform the indicated tasks.

STATS ERROR MESSAGES

STATS error messages are listed in Table 26.

STATS COMMAND SUMMARY

Table 27 contains a summary of commands. Table 28 contains a summary of interactive statements. The interactive statements are used in conjunction with the commands for defining the tasks that STATS is to perform.

Example 1

Create a display of performance data and output it to the line printer. (The report would be printed on the user's terminal if the !SET M:LO LP command were omitted.) No permanent file of performance data is generated.

```
!SET M:LO LP (RET)
!STATS (RET)
STATS B00 HERE
-ADD (RET)
 ENTER ITEMS TO BE ADDED
>ALL
>END (RET)
-DISPLAY (RET)
 ENTER INTERVAL IN MINUTES
>2 (RET)
 ENTER # OF INTERVALS
>10 (RET)
-END (RET)
!PRINT (RET)
```

The !PRINT command releases the listing to the printer.

Example 2.

Create snapshot files on-line.

```
!STATS (RET)
STATS B00 HERE
-FILE (RET)
 ENTER INTERVAL IN MINUTES
>5 (RET)
 ENTER # OF INTERVALS
>2 (RET)
 SNAPSHOT FILES OPENED
 RECORD #   0, 09:40 OCT 04,'71
 RECORD #   1, 09:45 OCT 04,'71
 RECORD #   2, 09:50 OCT 04,'71
 SNAPSHOT FILES CLOSED
-END (RET)
```

Example 3.

Run STATS as a ghost job. Every 15 minutes, snapshot records are added to the SNAPSHOT.:STATS and SSNAPSHOT.:STATS files. The job will "sleep" between times.

The job is initiated by the operator via the key-in

!GJOB STATS (NL)

The file GHOSTSI contains the following instructions to STATS:

```
*        STATS-B00 SLEEPING GHOST INPUT FILE TO CREATE SNAPSHOT FILES
*        KEY-IN ' GJOB STATS' AT THE BEGINNING OF EACH DAY
*                 AND AFTER EACH SYSTEM RESTART OR RECOVERY.
*        THE FOLLOWING FILES ARE CREATED AND/OR UPDATED
*                 SNAPSHOT.:STATS - CONSECUTIVE SNAPSHOT FILE
*                 SSNAPSHOT.:STATS - KEYED SNAPSHOT FILE
*                 GHOSTDO.:STATS - INTERACTIVE OUTPUT
*                 GHOSTLO.:STATS - REPORT OUTPUT
*        SNAPSHOT FILES MAY BE SAVED WEEKLY TO SAVE FILE SPACE
FILE
15
96
* 96 INTERVALS AT 15 MINUTES DURATION COVERS 24 HOURS.. EDIT TO SUIT
END
```

A message is typed on the operator's console to verify that the snapshot files are opened. At the conclusion, another message is typed on the operator's console to verify that the snapshot files are closed.

Table 26. STATS Error Messages

Message	Description
BUFFER TOO SMALL – REASSEMBLE	Not enough buffer space is available to copy all system performance data. A SYSGEN may be required to keep statistical data in low core. (This diagnostic indicates an unusual and perhaps unlikely case.)
BUILD LIST ERROR	The BUILD list number is missing or is not in the range 0-5.
CANNOT MAINTAIN INTERVAL	During operation of a DISPLAY command, the output device is unable to complete one display in time to print the next display. (The interval is too short.)
GHOSTDO FILE ERROR, M:DO – RESTART	STATS is being run as a ghost job and has created a disk file 'GHOSTDO' that is in error. The error is due either to lack of file space or to interference from another user. (Printed on operator's console.)
GHOSTLO FILE ERROR, M:LO – RESTART	STATS is being run as a ghost job and has created a disk file 'GHOSTLO' that is in error. The error is due either to lack of file space or to interference from another user. (Printed on operator's console.)

Table 26. STATS Error Messages (cont.)

Message	Description
GHOSTSI FILE ERROR, M:SI – RESTART	STATS is being run as a ghost job and the disk file 'GHOSTSI' is in error. Commands and responses to interactive questions and statements may only be input via GHOSTSI when STATS is run as a ghost job. (Printed on operator's console.)
ILLEGAL COMMAND	The user's response to a command level prompt (-) is not a legal command. (If this is a batch job, the commands may be out of sequence. Also, many commands do not allow abbreviations of the command name.)
NO PRINT FLAGS SET	A report has been requested but all print flags are 'OFF'.
NO SUCH NAME	An invalid name was used when entering items for an ADD or DROP command. Only the item names in Table 25 are valid. The standard display numbers are not valid.
PROCESSOR NOT FOUND IN MONITOR TABLE	The P:NAME table in the monitor does not contain the standard processors. This may indicate that the MONSTK for the system is different from the MONSTK used to load STATS.
SORRY, MONITOR ACCESS REQUIRES PRIV = 80	The user account must be reauthorized to allow the user to read monitor performance data.
SNAPSHOT FILE OUTPUT ERROR, F:2 – RESTART	An error has occurred during creation of a SNAPSHOT file entry. File space may have been exhausted or another user in the account may have interfered.
SSNAPSHOT FILE OUTPUT ERROR, F:3 – RESTART	An error has occurred during creation of a SSNAPSHOT file entry. It is also possible that file space allocated in the account may be exhausted.
UNEXPECTED CHANGE IN TIME – REPORT SKIPPED	This is either a user error, system error, or MONSTK problem since C:TIC and C:TINC should result in ever increasing values. A system recovery will reset these clock counters.
VIRTUAL PAGE NOT ALLOCATED, RE-ASSEMBLE	The number of core pages allocated in the user account is not sufficient.

Table 27. STATS Command Summary

Command	Description
-ADD ENTER ITEMS TO BE ADDED ≥item ≥item . . . ≥END	Turns on the print flags for the specified items. An item may be the name of any item in Table 25.
BREAK (i.e. (BRK))	Interrupts any current STATS operation and leaves control at the STATS command level. The PROCEED command is used to resume execution from the point of interrupt.

Table 27. STATS Command Summary (cont.)

Command	Description
B[UILD] n	Turns on the print flags for the standard display specified by n where n is a number in the range 0-5. See Table 25 for the numbers of the standard displays. BUILD 0 resets all the flags to off.
C[ONTROL]!	Displays all control parameters critical to system performance whether or not their print flags are on.
DISPLAY	Displays those items (from the group of items listed in Table 25) that have their print flags on. Snapshot files may be generated during DISPLAY command execution if the SNAPSHOT flag is on.
-DROP ENTER ITEMS TO BE DROPPED >item >item . . . >END	Turns off the print flags for the specified items. An item may be the name of any item in Table 25.
END	Causes an exit to TEL.
FILE	Creates snapshot records for the SNAPSHOT and SSNAPSHOT files. An on-line report may be generated simultaneously by turning the REPORT flag and the desired print flags on.
HELP	Briefly describes STATS commands and DCB usage.
LIST	Lists the STATS flags (other than standard display numbers) and their current status (on or off).
PROCEED	Continues STATS operation from the point at which the processor was interrupted by the BREAK key.
TIME	Lists the time and date.

Table 28. STATS Interactive Statements

Statement	Description of Response
ENTER INTERVAL IN MINUTES	Enter an integer (without a decimal point and without leading or embedded blanks). This number defines the length of the sample for the DISPLAY command.
ENTER ITEMS TO BE ADDED	Enter a list of display items, display groups, or print flags to be added for subsequent report format control.
ENTER ITEMS TO BE DROPPED	Enter a list of display items, display groups, or print flags to be dropped for subsequent report format control.
ENTER # OF INTERVALS	Enter an integer (without a decimal point and without leading or embedded blanks). This integer defines the number of intervals to be used for the DISPLAY command.

SUMMARY

The Summary processor provides a global view of system performance by formatting and displaying the statistical data collected by STATS. The input data for Summary is the SNAPSHOT and SSNAPSHOT files created by STATS. The output listings are generally large and therefore must be output to a file or on the line printer. The DCBs used by Summary are listed in Table 29.

The Summary processor allows the user to

1. Request a chronological listing of snapshot data for one or more display groups.

2. Specify a sort filter to remove undesired snapshots from the sample for subsequent reports.

3. Request filtered, sorted, and ordered listings of snapshot data for one or more display groups.

4. Request filtered, sorted, ordered, and averaged listings of snapshot data for one or more display groups.

5. Request means, minimums, maximums, and standard deviations for all display groups computed using the snapshots which pass the sort filter and a user specified intensity range. Correlation coefficients are included in this report that are estimates of the linear dependence between any pair of monitored variables.

Summary can be run in either the batch or on-line modes.

In the batch mode, Summary is called with the SUMMARY control command. The series of cards that follow require an anticipation of the interaction that occurs when Summary is run on-line. This point is explained in the fourth example of the section "Sample Summary Sessions". The rest of this discussion of Summary will assume that the processor is being run on-line.

Table 29. Summary DCB Usage

DCB	Function	On-line Default	Batch Default
F:2	Input consecutive snapshot file. May be SET or ASSIGNed.	fid=SNAPSHOT	fid=SNAPSHOT
F:3	Input keyed sorted snapshot file. May be SET or ASSIGNed.	fid=SSNAPSHOT	fid=SSNAPSHOT
M:DO†	Output interactive questions, statements, and error messages. If input through M:SI is from a file, it will be echoed through M:DO to provide a hard copy. May be SET or ASSIGNed.	user console	line printer
M:LO†	Output reports. May be SET or ASSIGNed, but should remain assigned to the line printer because of the report length.	line printer	line printer
M:SI	Input user responses to interactive questions. May be SET or ASSIGNed.	user console	after !SUMMARY card

†The line printer requires the AB flags in the account specification.

In the on-line mode, Summary is called by entering its name as a TEL command. Summary responds by typing SUMMARY HERE. Summary uses two prompt characters. The highest level prompt character is a single dash (-). The secondary prompt character is the 'greater than' character (>).

Example:

!SUMMARY (RET)

SUMMARY HERE

-

Control can be returned to TEL in three ways:

1. Depressing the BREAK key four times.
2. Depressing the CONTROL and Y keys.
3. Responding with an X to a yes/no type question from Summary.

SPECIAL COMMANDS

HELP The user may respond to any dash (-) prompt character with the HELP command. This command causes a listing to be output via M:DO that is designed to help the user in his interaction with the Summary processor. The listing that is output is shown in Figure 28. The format of the command is

HELP

LIST The user may respond to any dash (-) prompt character with the LIST command. This command causes a listing to be output via M:DO that lists the statistical item numbers and names and group names and numbers as shown in Figure 29. The format of the command is

LIST

```
SUMMARY IS A POST PROCESSOR FOR THE STATS PROCESSOR
REQUIRES SNAPSHOT AND SSNAPSHOT FILES
RESPOND TO QUESTIONS WITH ONE OF FOLLOWING
 Y(ES)  - TO SPECIFY GROUPS OR POSITIVE RESPONSE
 N(O) OR CARRIAGE RETURN - TO SKIP TO NEXT QUESTION
 ALL    - TO PROCESS ALL STATISTICAL GROUPS
 HELP   - TO GET THIS LISTING
 LIST   - TO GET LIST OF STATISTICS BY NUMBER
 X TO EXIT
 END TO TERMINATE GROUP LISTS OR TO EXIT TO TEL)
 *  COMMENT LINE
 DCB USAGE
M:SI - INTERACTIVE INPUT
M:DO - INTERACTIVE OUTPUT
M:LO - REPORT OUTPUT
F:2  - CONSECUTIVE SNAPSHOT FILE
F:3  - KEYED SNAPSHOT FILE
 RECOMMENDED STANDARD PROCEDURE FOLLOWS
YES        (CHRONOLOGICAL SUMMARIES   )
2          (CPU)
6          (I/O)
NO         (MORE GROUPS   )
YES        (SPECIFY SORT FILTER   )
34,0,10          (LESS THAN 10% CPU IDLE)
END        (TERMINATES FILTER LIST)
NO         (SORTED SUMMARIES   )
ALL        (SORTED AVERAGES   )
YES        (CORRELATION ANALYSIS   )
           (DEFAULT)
           (DEFAULT)
NO         (MORE SORTS   )
```

Figure 28. HELP Command Listing

```
LISTING OF STATISTICAL GROUPS

   0 - PARAM
 1 HOUR:MINUTES
 2 INTERVAL IN MIN
 3 MAX # BATCH
 4 MAX # ONLINE
 5 AVERAGE BATCH K
 6 AVERAGE ONLINE K
 7 # CHAR TERM BLK
 8 # CHAR TERM UBLK
 9 QMIN          MSEC
10 SQUAN         MSEC
11 AVE BATCH QUAN
12 MSEC ONLINE QUAN
13 COMPUTE Q SHARE
14

   1 - SUMMARY
15 BATCH STREAMS
16 ONLINE USERS
17 % CPU/BATCH USER
18 % CPU/ONLINE USE
19 BATCH EXEC/SERV
20 ONLINE EXEC/SERV
21 CPU MSEC PER I/O
22 ONLINE TIME MIX
23 ONLINE INTENSITY
24 ONLINE TASKS/MIN
25 % INTERACTIVE
26 90% RESPONSE   MS
27 TURNAROUND    SEC
28 ETMF

   2 - CPU
29 BATCH EXEC
30 BATCH SERV
31 ONLINE EXEC
32 ONLINE SERV
33 MONITOR SERV
34 IDLE
35 SWAP WAIT
36 I/O WAIT
37 I/O&SWAP WAIT
38 TOTAL

   3 - BATCH
39
40
41
42
43   processor names dependent upon
44   installation
45
46
47
48
49 USER
50 SHARED
```

Figure 29. LIST Command Listing

```
   4 - ONLINE
51
52
53
54
55   processor names dependent upon
56   installation
57
58
59
60
61 USER
62 SHARED

   5 - USERS
63 BASIC
64 DELTA
65 EDIT
66 FDP
67 FORTRAN
68 FORTLIB
69 LINK
70 LOADER
71 METASYM
72 PCL
73 TEXT
74 IN CORE

   6 - I/O
75 SERVICE REQ
76 INTERACTIONS
77 CHAR IN
78 CHAR OUT
79 TERM WRITES
80 I/O ACCESSES
81 SYMBIONT
82 IN SWAPS
83 OUT SWAPS

   7 - TASK
84 INTERACT
85 THINK-TY
86 TURNARND
87 COMPLETE
88 RESPONSE     MSEC
89 CPU          MSEC

   8 - QUEUE
90 LOGGED
91 LOGGING
92 BATCH
93 GHOST
94 TERM IN
95 TERM OUT
96 COMPUTE
97 COMP BND
98 I/O
99 SLEEP
```

Figure 29. LIST Command Listing (cont.)

INTERACTION WITH SUMMARY

The Summary processor interacts with a user through a series of interactive questions and statements. The user's responses to these questions and statements define the tasks required of Summary. There are six major questions asked by Summary. They are asked in the following order:

1. CHRONOLOGICAL SUMMARIES?
2. SPECIFY SORT FILTER?
3. SORTED SUMMARIES?
4. SORTED AVERAGES?
5. CORRELATION ANALYSIS?
6. MORE SORTS?

When the user's response to one of these questions is affirmative (YES), Summary begins the designated task. Secondary questions or statements from Summary and user responses then define the task more specifically. (All Summary questions and appropriate responses and all tables at the end of the chapter.) Figure 30 provides a flowchart of Summary's interaction with the user. (The secondary type questions and statements are not indicated on that chart.)

TASKS SUMMARY PERFORMS

CHRONOLOGICAL SUMMARIES

A chronological report of system performance statistics may include from one to eleven statistical groups of data. The ten possible statistical groups are listed in Table 30. The STATS flag(s) that is associated with each statistical group is indicated in parentheses.

When the question 'CHRONOLOGICAL SUMMARIES?' is asked, the user's response may be

ALL which requests that all statistical groups be displayed. After the report, the next major question will be output.

YES which requests that one or more statistical groups be displayed.

{X END} which stops the Summary processor and returns control to TEL.

other which causes the next major question to be asked and the chronological summaries task to be skipped.

If the response is YES, Summary will then type the statement

ENTER ALL, OR GROUP #0-9

The user's response to this statement may be

ALL which requests that all statistical groups be displayed.

n which specifies the number (0-9) of a statistical group to be displayed. If the number is outside the range 0-9, the statement is repeated.

{X END} which stops the Summary processor and returns control to TEL.

other which causes the next major question (SPECIFY SORT FILTER?) to be asked.

If the number of a statistical group is specified, that group is displayed and then the following question is asked:

ANOTHER GROUP?

The response to this question may be

ALL which requests that all statistical groups be displayed.

YES which requests that one or more statistical groups be displayed and leads to the 'ENTER ALL, OR GROUP #0-9' statement.

0-10 which also requests that all statistical groups be displayed.

other which causes the next major question (SPECIFY SORT FILTER?) to be asked.

The question 'ANOTHER GROUP?' enables the user to select another statistical group for display and will be used repeatedly until the user indicates that Summary should go on to the next major question or that control should be returned to TEL.

The snapshot data for the chronological summaries is read from the consecutive 'SNAPSHOT' file via the F:2 DCB in the order in which the snapshots were created. The SNAPSHOT file may contain snapshots from more than one history file. This can happen because

1. More than one history file may be processed by STATS in one run.
2. Several copies of previous SNAPSHOT files may be merged into one file.

The statistics are averaged at the end of each snapshot file.

The chronological listing of group '0' is different from the other groups in that only the snapshots in which control parameters that have changed are listed. Since the parameters remain relatively constant, this eliminates the needless printing of many lines of repetitious data.

There is one line of output for each snapshot record of each statistical group. Therefore the listings are long and it is anticipated that chronological summaries will be most useful for daily and weekly reports.

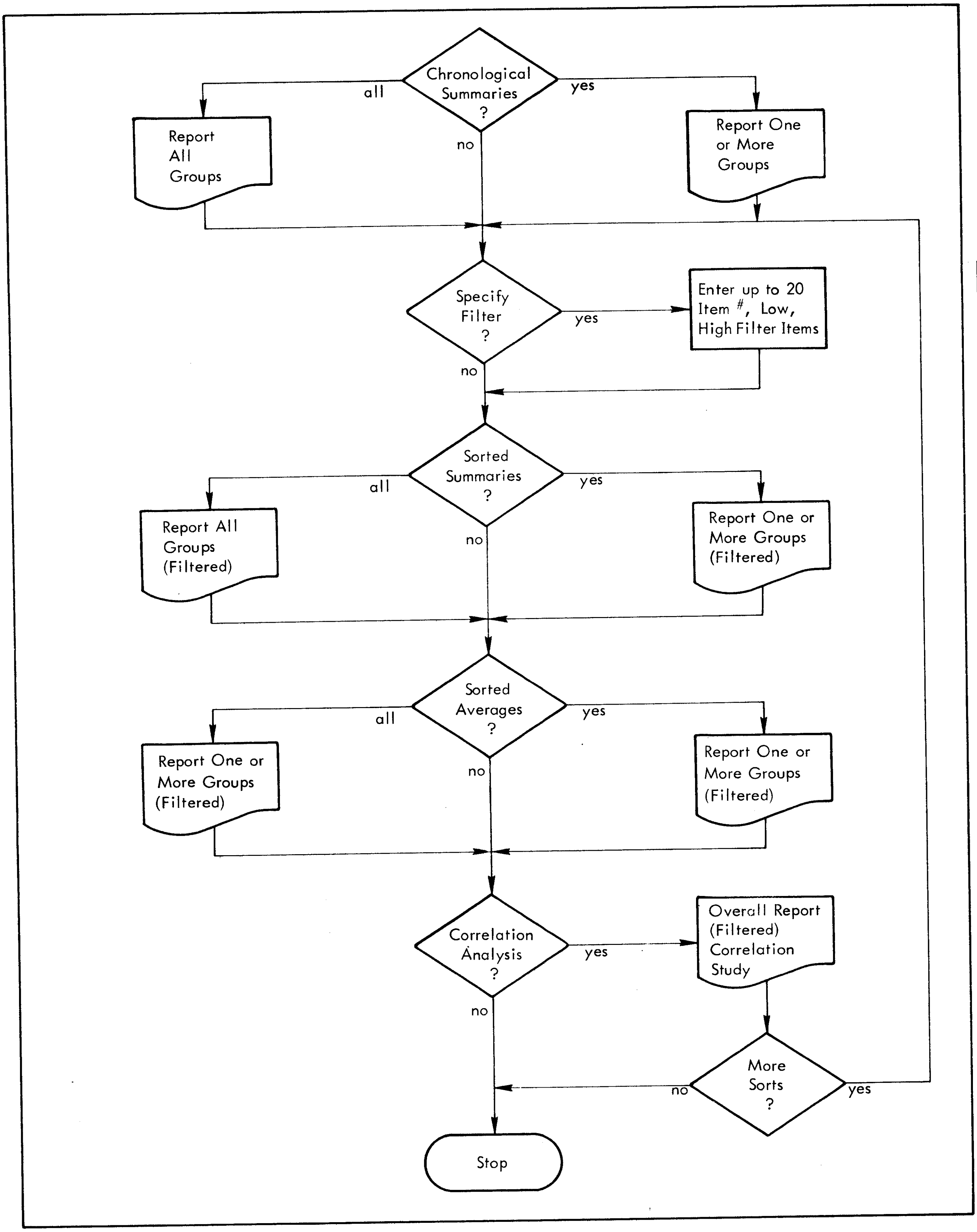


Figure 30. Flow Chart of the Summary Processor Interactive Questions

Table 30. Summary Statistical Groups

Group No.	Description
0	Selected parameters. (PARAM)
1	Summary statistics. (SUMMARY)
2	CPU utilization - execution, service, idle, and swap wait. (CPU)
3	Percent of CPU time per monitored processor for batch users. (BATCH)
4	Percent of CPU time per monitored processor for on-line users. (ONLINE)
5	Number of users per processor. (USERS)
6	I/O rates. (I/O)
7	Task statistics and system response. (TASK)
8	Queue statistics. (QUEUE)
9	System histograms for statistics and swap statistics. (SYSTEM and SWAP)

SORT FILTER SPECIFICATION

It is extremely useful to be able to reject snapshots which fail to meet certain criteria. The sort filter allows the user to specify up to 20 variables with high and low limits. All snapshots which fail to meet these limits are rejected from the display samples used in creating reports.

The user is asked if he would like to specify sort filters by the question 'SPECIFY SORT FILTERS ?'. His response may be one of the following:

YES which indicates that he would like to specify sort filters.

{X / END} which stops the Summary processor and returns control to TEL.

other which causes the next major question to be asked.

If the response is YES, the message 'ENTER ITEM#,LOW LIMIT,HIGH LIMIT' is typed. The response to this message may be

n1,n2,n3 which specifies an item number (n1), a low limit (n2), and a high limit (n3). N1 must be an integer in the range one to the maximum number of monitored items. N2 and n3 are numbers with or without decimal points. If n3 is smaller than n2, the input line will be rejected and the 'ENTER...' statement will be repeated. Twenty parameters and associated limits may be entered. (The procedure for selecting parameters and their limits is discussed shortly.)

null which terminates the list of items and limits. The next main question is asked.

other which may lead to a FORTRAN run-time error.

The following procedure should be used to select sort parameters:

Use the LIST command to identify item numbers. Then pick the filter items by number and choose a high and low limit for each item. Typically one chooses a minimum limit that is at least one standard deviation lower than the mean and a maximum limit that is at least one standard deviation above the mean. The report can be used to obtain the statistics needed for a logical choice for high and low limits. The time of day, the interval length, the amount of idle, and compute queue sharing are items which are useful in selecting a sort filter. Of course, any limits that are appropriate for the needs of the particular installation may be chosen.

SORTED SUMMARIES

It is useful to compare snapshot data when the snapshots have been sorted by estimated user intensity, ordered by the number of users within an intensity range, and filtered by rejecting all snapshots that fail to meet the sort filter limits. The STATS processor sorts and orders the snapshot data by writing keyed records that have keys defined by the intensity, number of logged users, and the date and time and stores the results in the SSNAPSHOT file. The PCL processor may be used to merge previous copies of SSNAPSHOT into one sorted and ordered file. Each sorted snapshot record has a unique key.

The snapshot data for the sorted summaries is read from the SSNAPSHOT file via the F:3 DCB sequentially. The sort filter criteria is applied and only those snapshots that pass the criteria are accepted for display. The reports lists one or more sophisticated groups with an average listed for each range of intensity. The low intensity snapshots tend to be for environments that require little CPU time; the high intensity snapshots tend to be for environments that require a lot of CPU time. Many statistical trends become apparent in sorted summaries which are lost in chronological summaries.

The following responses may be given to the question 'SORTED SUMMARIES ?':

ALL which requests that all statistical groups be displayed after filtering, sorting, and ordering. After the report the next major question (SORTED AVERAGES ?) will be output.

YES which requests that one or more statistical groups be displayed after filtering, sorting, and ordering. This response causes the 'ENTER ALL, OR GROUP #0-9' message to be printed. One or more groups may be specified as in chronological summaries.

$\begin{Bmatrix} \text{X} \\ \text{END} \end{Bmatrix}$ which stops the Summary processor and returns control to TEL.

other which causes the next major question (SORTED AVERAGES?) to be asked and the sorted summaries to be skipped.

SORTED AVERAGES

When there are a large number of snapshots, it is useful to average snapshot statistics within the same intensity block and with the same number of users. When sorted averages are requested, the SSNAPSHOT file provides the data and is input via the F:3 DCB.

Summary processes the SSNAPSHOT file sequentially and applies the sort filter criteria before accepting the snapshot data for the sample. The report lists the averages for each number of users logged within each user intensity group. Each intensity group is averaged as a group just as in sorted summaries.

The user is asked if he would like to have sorted averages by the question 'SORTED AVERAGES?'. The user's response may be one of the following:

ALL which requests that all statistical groups be displayed after filtering, sorting, ordering, and averaging. After the report, the next major question will be output.

YES which requests that one or more statistical groups be displayed after filtering, sorting, ordering, and averaging. This response causes the 'ENTER ALL, OR GROUP #0-9' message to be printed. One or more groups may be specified as in chronological summaries.

$\begin{Bmatrix} \text{X} \\ \text{END} \end{Bmatrix}$ which stops the Summary processor and returns control to TEL.

other which causes the next major question (CORRELATION ANALYSIS?) to be asked and the sorted averages task to be skipped.

CORRELATION ANALYSIS

When the question 'CORRELATION ANALYSIS?' is asked, a YES response begins the generation of a comprehensive report that statistically summarizes the snapshots that pass the sort filter. Any response other than YES stops the Summary processor and returns control to TEL.

If the response is YES, the following statement prints:

ENTER CORRELATION THRESHOLD. (0.0 to 0.99)

This allows the user to specify a threshold which is used to suppress small correlation coefficients from the listing. The user's response may be a threshold value or may be null. If the response is null, the default threshold value (.1) is used. The threshold value is retyped by STATS so that the user may verify it.

After a response to the above statement is entered, the following statement prints:

ENTER INTENSITY RANGE. (XLOW, XHIGH)

This allows the user to limit the number of snapshots in the sample by specifying the intensity range. The user may enter two intensities separated by a comma or take the default of 200, 2200 by entering a null response. The intensity range is retyped by STATS so that the user may verify it. The correlation analysis report is then generated.

The correlation analysis report has the following organization:

1. Header stating the number of accepted snapshots in the sample.

2. List of filter parameters with number of rejections listed for each limit.

3. List of all monitored items and associated overall statistics (name, mean, minimum, maximum, and standard deviation). The statistics are gathered for all of the snapshots which passed the filter and intensity restrictions. Figure 31 shows the format of the overall summary.

4. Covariance matrix printed in sections, 20 columns at a time. Each matrix entry is a correlation coefficient which is a measure of the linear dependence between the monitored items defined by the row and the column for each coefficient. The matrix is symmetric about the diagonal. Each entry on the diagonal is zero or 1.0 by definition. Every element must be in the range -1.0 to +1.0. Low valued correlation coefficients do not guarantee independence. Negative coefficients indicate inverse linear relationships. Positive coefficients indicate a linear relationship. Nonlinear relationships may exist with small correlation coefficients. Figure 32 shows a section of the 99 x 99 matrix.

5. Histograms describing task statistics and swap statistics for the overall sample. These are obtained from the data in the histogram listings from the STATS processor. (See Figure 33.)

```
ØVERALL SUMMARY ØF    14 SNAPSHØTS SELECTED FRØM    17
INTENSITY RANGE FRØM          0. TØ 000000000.
FILTER RETRICTIØNS FØR SAMPLE
   #                             LØW       HIGH      # LØW      # HIGH
  34 IDLE                         .0       10.0         .0         3.0
     0 - PARAM
   #                          MEAN       MIN       MAX      STD.DEV.
   1 HØUR:MINUTES
   2 INTERVAL IN MIN
   3 MAX # BATCH
   4 MAX # ØNLINE
   5 AVERAGE BATCH K
   6 AVERAGE ØNLINE K
   7 # CHAR TERM BLK
   8 # CHAR TERM UBLK
   9 QMIN           MSEC
  10 SQUAN          MSEC
  11 AVE BATCH QUAN
  12 MSEC ØNLINE QUAN
  13 CØMPUTE Q SHARE
  14
     1 - SUMMARY
   #                          MEAN       MIN       MAX      STD.DEV.
  15 BATCH STREAMS
  16 ØNLINE USERS
  17 % CPU/BATCH USER
  18 % CPU/ØNLINE USE
  19 BATCH EXEC/SERV
  20 ØNLINE EXEC/SERV
  21 CPU MSEC PER I/Ø
  22 ØNLINE TIME MIX
  23 ØNLINE INTENSITY
  24 ØNLINE TASKS/MIN
  25 % INTERACTIVE
  26 90% RESPØNSE  MS
  27 TURNARØUND   SEC
  28 ETMF
     2 - CPU
   #                          MEAN       MIN       MAX      STD.DEV.
  29 BATCH EXEC
  30 BATCH SERV
  31 ØNLINE EXEC
  32 ØNLINE SERV
  33 MØNITØR SERV
  34 IDLE
  35 SWAP WAIT
  36 I/Ø WAIT
  37 I/Ø&SWAP WAIT
  38 TØTAL
     3 - BATCH
   #                          MEAN       MIN       MAX      STD.DEV.
  39 LINK
  40 DELTA
  41 :P00
  42 :P11
  43 LDEV
  44 EDIT
  45 PCL
  46 BASIC
  47 METASYM
  48 LØADER
  49 USER
  50 SHARED
```

Figure 31. Format of Overall Summary

```
CORRELATION ANALYSIS OF    14 SELECTED FROM  17
CORRELATION COEFFICIENTS <  .10 SUPPRESSED.
INTENSITY RANGE FROM         0. TO 000000000.
                  <--------------------- 0 - PARAM ------------------><-------- 1 - SUMMARY---
   0 - PARAM        1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20
 1 HOUR:MINUTES
 2 INTERVAL IN MIN
 3 MAX # BATCH
 4 MAX # ONLINE
 5 AVERAGE BATCH K
 6 AVERAGE ONLINE K
 7 # CHAR TERM BLK
 8 # CHAR TERM UBLK
 9 QMIN          MSEC
10 SQUAN         MSEC
11 AVE BATCH QUAN
12 MSEC ONLINE QUAN
13 COMPUTE Q SHARE
14
                  <--------------------- 0 - PARAM ------------------><-------- 1 - SUMMARY---
   1 - SUMMARY      1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20
15 BATCH STREAMS
16 ONLINE USERS
17 % CPU/BATCH USER
18 % CPU/ONLINE USE
19 BATCH EXEC/SERV
20 ONLINE EXEC/SERV
21 CPU MSEC PER I/O
22 ONLINE TIME MIX
23 ONLINE INTENSITY
24 ONLINE TASKS/MIN
25 % INTERACTIVE
26 90% RESPONSE  MS
27 TURNAROUND   SEC
28 ETMF
                  <--------------------- 0 - PARAM ------------------><-------- 1 - SUMMARY---
   2 - CPU          1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20
29 BATCH EXEC
30 BATCH SERV
31 ONLINE EXEC
32 ONLINE SERV
33 MONITOR SERV
34 IDLE
35 SWAP WAIT
36 I/O WAIT
37 I/O&SWAP WAIT
38 TOTAL
```

(correlation coefficients $-1 \leq \rho ij \leq +1$. Values close to +1 indicate positive correlation and possible positive linear correlation. Values close to zero indicate possible linear independence. Negative values close to -1 indicate possible inverse linear correlation.)

Figure 32. A Section of the Covariance Matrix

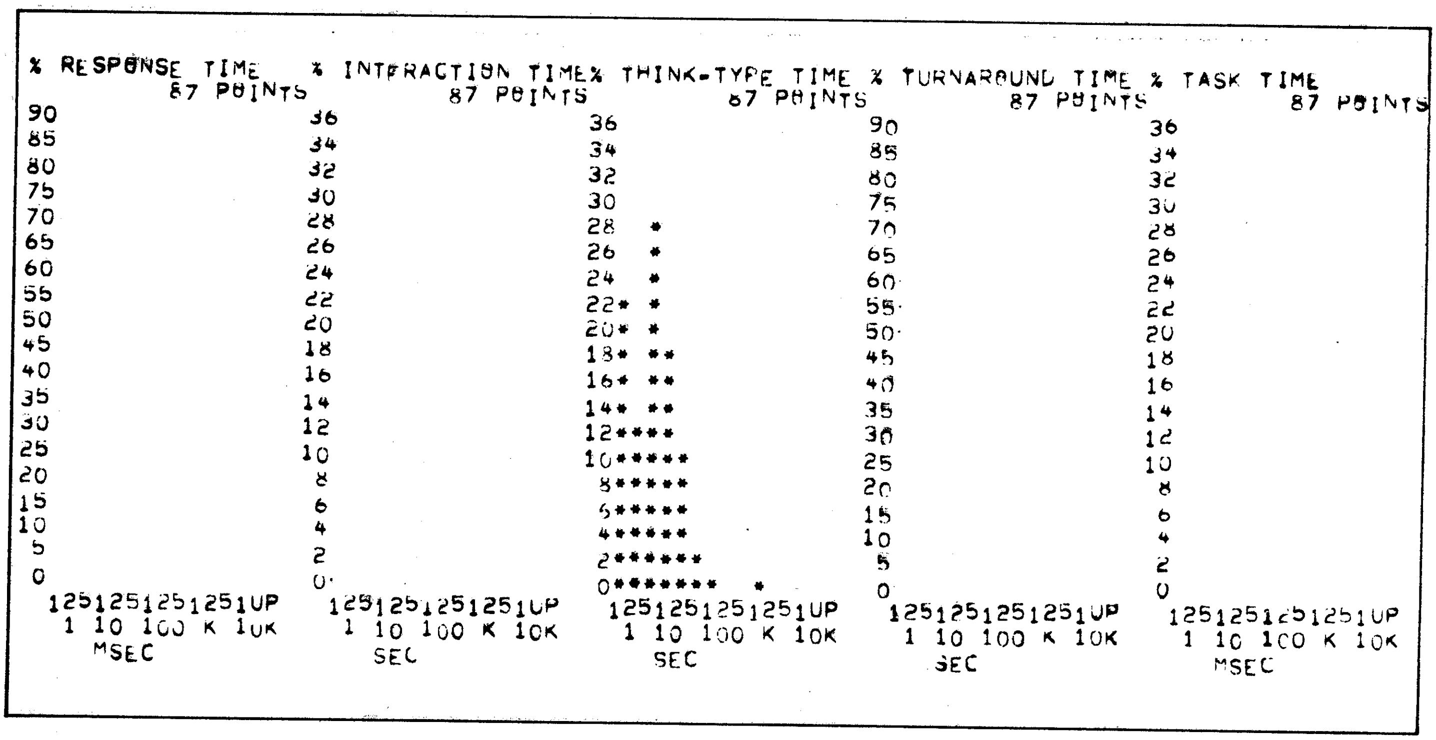


Figure 33. Task and Swap Statistics

SAMPLE SUMMARY SESSIONS

The following are sample Summary sessions that perform the indicated tasks:

1. Report all available statistics without filtering.

```
!SET M:LO LP

!SUMMARY
 SUMMARY HERE
 CHRONOLOGICAL SUMMARIES?
-ALL
 SPECIFY SORT FILTER?
-NO
 SORTED SUMMARIES?
-ALL

 SORTED AVERAGES?
-ALL
 CORRELATION ANALYSIS?
-YES
 ENTER CORRELATION THRESHOLD (0.0 TO .99)
>
 THRESHOLD = .00
 ENTER INTENSITY RANGE. (XLOW,XHIGH)
>
 INTENSITY RANGE FROM 0. TO 10000.
 100 SNAPSHOTS WERE SELECTED FROM 100
 MORE SORTS?
-NO
 *STOP* 0

!PRINT
```

2. Set filter limits to restrict the control parameters within limits and produce a sorted, ordered, and filtered summary of all statistical groups.

```
!SET M:LO LP

!SUMMARY
 SUMMARY HERE
 CHRONOLOGICAL SUMMARIES?
-NO
 SPECIFY SORT FILTER?
-YES
 ENTER ITEM#,LOW LIMIT,HIGH LIMIT
>7,40,60
 7 % BATCH BIAS                 40.0      60.0
>10,30,50
 10 MIN QUANTUM                 30.0      50.0
>11,300,500
 11 COMPUTE QUANTUM            300.0     500.0
>
 SORTED SUMMARIES?
-NO
 SORTED AVERAGES?
-ALL
 CORRELATION ANALYSIS?
-X
 *STOP* 0

!PRINT
```

3. Report chronological statistics for groups 1, 2, and 5. Note that an integer response to the 'ANOTHER

GROUP?' question skips the 'ENTER ALL, OR GROUP #0-9' interaction.

```
!SET M:LO LP

!SUMMARY
 SUMMARY HERE
 CHRONOLOGICAL SUMMARIES?
-YES
 ENTER ALL, OR GROUP # 0-9
 1
 ANOTHER GROUP?
 2
 ANOTHER GROUP?
 YES
 ENTER ALL, OR GROUP # 0-9
 5
 ANOTHER GROUP?

 SPECIFY SORT FILTER?
-X
 *STOP* 0

!PRINT
```

4. Run complete summary as a batch job. Note that when running Summary as a batch job, the user must anticipate each question or statement for which Summary will expect a response. The job is essentially a list of the responses.

```
!SUMMARY
ALL        (CHRONOLOGICAL SUMMARIES?)
NO         (SPECIFY SORT FILTER?)
ALL        (SORTED SUMMARIES?)
ALL        (SORTED AVERAGES?)
YES        (CORRELATION ANALYSIS?)
.3         (ENTER CORRELATION THRESHOLD.
               (0.0 TO .99))
200,2200   (ENTER INTENSITY RANGE.
               (XLOW, XHIGH))
NO         (MORE SORTS?)
!EOD
```

5. Get overall statistical summary without filtering. Summary is run is a batch job.

```
!SUMMARY
NO         (CHRONOLOGICAL SUMMARIES?)
NO         (SORT FILTER?)
NO         (SORTED SUMMARIES?)
NO         (SORTED AVERAGES?)
YES        (CORRELATION ANALYSIS?)
blank      (ENTER CORRELATION THRESHOLD.
               (0.0 TO .99))
blank      (ENTER INTENSITY RANGE.
               (XLOW, XHIGH))
NO         (MORE SORTS?)
```

SUMMARY ERROR MESSAGES

If Summary tries to open SNAPSHOT or SSNAPSHOT and the file does not exist, the following message is output.

4603 CAN'T OPEN FOR READ: FILE DOESN'T EXIST

SUMMARY INTERACTIVE QUESTIONS AND STATEMENTS

The questions and statements with which the Summary processor interacts with the user are summarized in Tables 31 and 32 respectively.

Table 31. Summary Interactive Questions

Questions	User Response
ANOTHER GROUP?	YES -requests that one or more statistical groups be displayed. Leads to the 'ENTER ALL, OR GROUP # 0-9' statement. {ALL, 0-9} -requests that all statistical groups be displayed. other -leads to the next major interactive question.
CHRONOLOGICAL SUMMARIES?	ALL - requests that all statistical groups be displayed chronologically. YES -requests that one or more statistical groups be displayed chronologically. Leads to the 'ENTER ALL, OR GROUP # 0-9' statement. X or END -stops the Summary processor and returns control to TEL. other -causes the next major question to be asked and the chronological summaries task to be skipped.

Table 31. Summary Interactive Questions (cont.)

Questions	User Response
CORRELATION ANALYSIS?	YES -leads to the 'ENTER CORRELATION THRESHOLD' statement and ultimately to an overall report that includes the correlation matrix. other -stops the Summary processor and returns control to TEL.
MORE SORTS?	YES -leads to the 'SPECIFY SORT FILTER?' question for another filtered sort. other -stops the Summary processor and returns control to TEL.
SORTED AVERAGES?	ALL -requests that all statistical groups be displayed after filtering, sorting ordering, and averaging. YES -requests that one or more statistical groups be displayed after filtering, sorting, ordering, and averaging. X or END -stops the Summary processor and returns control to TEL. other -causes the next major question to be asked and the sorted averages task to be skipped.
SORTED SUMMARIES?	ALL -requests that all statistical groups be displayed after filtering, sorting, and ordering. YES -requests that one or more statistical groups be displayed after filtering, sorting, and ordering. X or END -stops the Summary processor and returns control to TEL. other -causes the next major question to be asked.
SPECIFY SORT FILTER?	YES -leads to the 'ENTER ITEM #, LOW LIMIT, HIGH LIMIT' statement for user definition of filter limits. X or END -stops the Summary processor and returns control to TEL. other -causes the next major question to be asked.

Table 32. Summary Interactive Statements

Statement	User Response
ENTER ALL, OR GROUP # 0-9	ALL -has the same meaning as an ALL answer to the question asked just before this statement. n -specifies the number (0-9) of a statistical group to be displayed. If the number is outside the range 0-9, the statement is repeated. X or END -stops the Summary processor and returns control to TEL. other -causes the next major question to be asked.

Table 32. Summary Interactive Statements (cont.)

Statement	User Response	
ENTER CORRELATION THRESHOLD (0.0 TO .99)	n	-specifies a number with a decimal point within the range 0.0 to .99. If the number is outside the range, the statement is repeated.
	null	-specifies a threshold of .1 by default.
	other	-may lead to a FORTRAN run-time error.
ENTER INTENSITY RANGE. (XLOW,XHIGH)	n1,n2	-specifies an intensity range, where n1 and n2 are numbers with or without decimal points and separated by a comma. If n2 is smaller than n1, the statement will be repeated.
	null	-specifies a range of 0 to 10000 by default.
	other	-may lead to a FORTRAN run-time error.
ENTER ITEM#,LOW LIMIT,HIGH LIMIT	n1,n2,n3	-specifies an item number (n1), a low limit (n2), and a high limit (n3). N1 must be an integer in the range one to the maximum number of monitored items. N2 and n3 are numbers with or without decimal points. If n3 is smaller then n2, the statement will be repeated. Twenty parameters and associated limits may be entered.
	null	-terminates the list of items and limits.
	other	-may lead to a FORTRAN run-time error.

7. SHARED PROCESSOR FACILITIES

INTRODUCTION

This chapter describes the shared processor facilities of CP-V. These facilities permit the sharing of the code for compilers, assemblers, command language processors, debuggers, libraries, and other programs among all simultaneous users.

Shared processors are not limited to programs provided by Xerox. The facilities may be effectively used whenever a program has a high probability of common usage. Service bureaus, for example, may use the mechanism for proprietary packages. Corporate installations may use the mechanism for programs with a high use frequency.

Most programs may be established as shared processors by naming them at SYSGEN time. This causes the file copy of the program from the :SYS account to be written on the swapping disk during system initialization. The program is then available through high-speed swapping I/O.

The file copy of the program is retained for recovery purposes and may be copied to another account and run as an unshared program under Delta for development and debugging purposes. If the load module in the :SYS account is replaced, the shared copy of the program on the swapping disk is updated to the newer version in the event of a system recovery.

To qualify as a shared processor, a program must meet certain requirements. These requirements are outlined in the remainder of this chapter. The most stringent requirement relates to the single overlay level that is described in the section below titled "Overlay Restrictions".

PUBLIC PROGRAMS

A program whose load module is in the :SYS account is a public program in the sense that it may be called either by a control card containing the ! symbol and the program name, or by an entry of the program name in response to a TEL prompt (!) for commands. Each user of a public program has his own copy of the program.

SHARED PROGRAMS

Shared programs are called in the same manner as public programs. However, each user of a shared program has his own copy of only the data and DCB portion of that program; the procedure portion is shared by all users associated with the shared program.

There are four distinct kinds of shared programs:

1. Ordinary shared processors.

2. Special shared processors.

3. Shared debuggers.

4. Public libraries.

All shared processors must be built by the batch loader. Memory allocation for an ordinary shared processor is shown in Figure 2. Ordinary shared processors occupy the same virtual memory as user programs and may not be associated with them.

Special shared processors, shared debuggers and public libraries occupy (and are overlayed in) the special processor area. Figure 34 shows the virtual memory allocation for shared programs that are biased within the special processor area. Shared debuggers may be associated only with user programs; they may not be associated with any other shared processors. Public libraries may be associated with user programs or ordinary shared processors; a public library may not be associated with a special shared processor. Note that both a shared debugger and a core library may be concurrently associated with a user program. This is possible because the procedure portion of the debugger and the library may be overlayed in the special processor area.

LOG-ON CONNECTION

Commonly used programs, such as BASIC, may be called automatically by LOGON. The name of the program to be called, which may be either a shared or public program

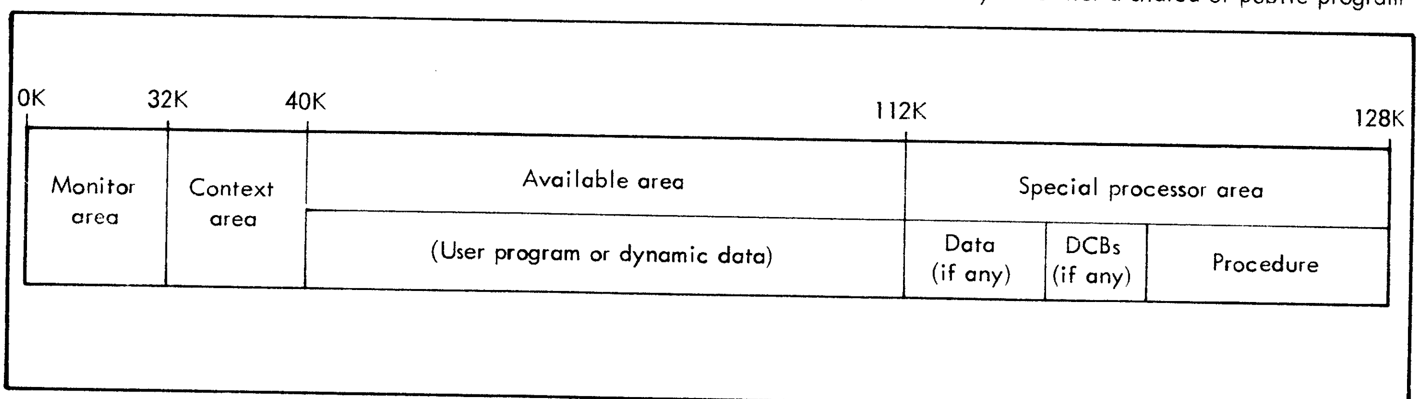


Figure 34. Special Processors – Virtual Memory

from any accessible account, is established in the user's log-on record by Super. LOGON calls the named program for the user following a successful log-on.

SHARED PROCESSOR PROGRAMMING

The programming of shared processors may require certain information about the CP-V monitor. This information is outlined below.

FIXED MONITOR LOCATIONS

For certain purposes, such as the choice of an effective core allocation technique, it is desirable for processors and other programs to be able to identify the monitor in operation, certain critical locations of the monitor, and the location of job information table (JIT). This is accomplished by having locations 2A, 2B, and 4F common to all Xerox monitors. Figure 35 illustrates the contents of these locations.

Location 2A contains a flag that differentiates between an initial boot (nonzero) and a recovery boot (zero).

Location 2B contains three items:

1. Monitor – This field contains the code number of the monitor. The codes are as follows:

Code	Monitor
0	None or indeterminate
1	BCM
2	RBM
3	RBM-2
4	BPM
5	BTM/BPM
6	UTS
7	CP-V
8-F	Reserved for future use

2. Version – This is the version code of the monitor and is coded to correspond to the common designation for versions. The alphabetic count of the version designation is the high-order part of the code and the version number is the low-order part. For example, A00 is coded X'10' and D02 is coded X'42'.

3. Parameters – The bits in this field are used to indicate suboptions of the monitor. They are meaningful only in relation to a particular monitor. However, the following assignments have been made for BPM, BTM, and CP-V.

Bit	Meaning if Set
31	Symbiont routines included.
30	Remote processing routines included.
29	Real-time routines included.
28	Unused.
27	Reserved for Data Management System.
26	Reserved.
25	Machine is Sigma 6 or 7 (zero for Sigma 5 or 9).
24	Machine is Sigma 9.

Location 4F contains the virtual JIT address right-justified.

JOB INFORMATION TABLE (JIT)

For each active job, the system maintains an in-core record (job information table) that allows the job to be scheduled and swapped. This job information table (JIT) is the first page of each job, both in core and on the swapping disk, and contains accounting information, memory map, swap storage, addresses, and other information for the job that

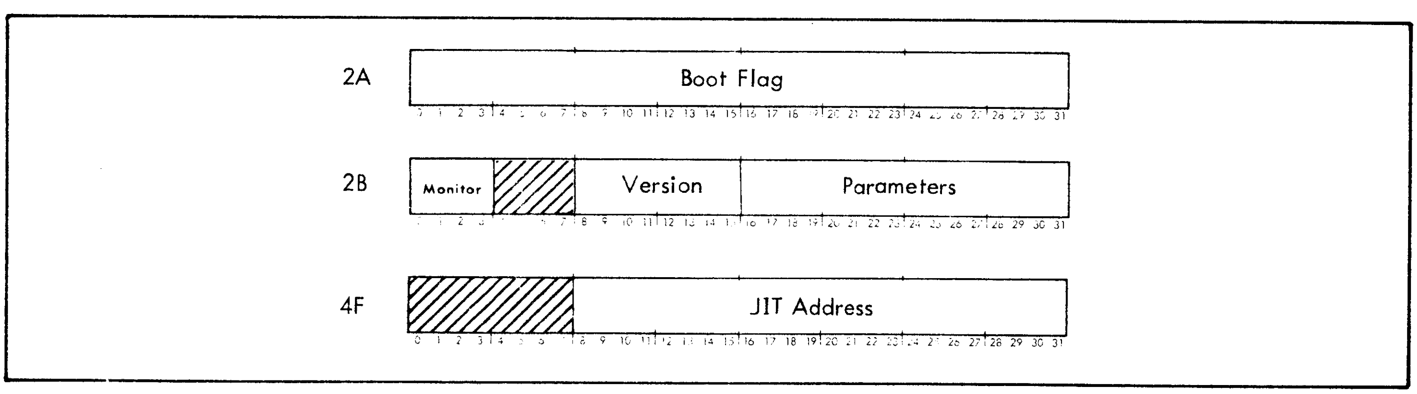


Figure 35. Locations Common to All Monitors

may be of use to a processor.[†] In order to reference these values, the processor should REF the required symbol and then specify that :J0, the JIT definition package, be loaded along with the processor. The entire JIT is available on a read-only basis to all programs including processors. Contents that are particularly useful to processors are given in Table 33.

Table 33. Partial Contents of JIT

Location	Size	Contents
J:JIT		
(bit 0)	1 bit	Set if the job is on-line and reset if the job is batch.
(bit 1)	1 bit	Set if the job is a ghost job. For example, the meaning of bits 0 and 1 is as follows: 00 batch job 01 ghost job 10 on-line
(bits 16-31)	halfword	Job identification number that is guaranteed to be unique to each currently executing job.
JB:LPP	byte	Number of printable lines per page (COC).
JB:LC	byte	Current print line number (COC).
JOPT	word	Peripheral usage flags set by TEL (see section titled "TEL Scan").
J:TELBUF	20 words	Image of the command line received by TEL.
J:USER	2 words	On doubleword boundary for any use by installation.
M:UC	22 words	Console I/O DCB.

[†]With respect to accounting, only shared processors are processors, i.e., time spent compiling a COBOL program is accounted under "user time" while time spent in FORTRAN, PCL, etc., is considered "processor time".

MEMORY CONTROL

No special memory restrictions apply to programs operating as shared processors. In CP-V, as in any other time-shared or multiprogrammed system, prudent use of memory can substantially improve system throughput. Requests for all available memory should be avoided. A request for enough memory to cover typical processing should be made initially, then a request for additional memory should be made during processing if the need arises. Memory should be returned to the system at major changes of control, but the frequent acquisition and release of memory will increase system overhead out of proportion to the gain.

OVERLAY RESTRICTIONS

Any processor intended for shared use may be created and debugged as an ordinary program. It may be coded in assembly language and debugged under Delta or created in FORTRAN and debugged with FDP. To qualify for inclusion as a shared processor, it must be coded within the following restrictions:

1. Shared processors are allowed only one level of overlay. There is no restriction on the number of overlays but only one of them can be associated at a time.
2. Data cannot be included in overlays; it must be in the processor root.
3. Overlay names are restricted to seven characters or less.
4. All parts of an overlay disappear from core when another overlay is called. (Portions of a previously used overlay are not available when a shorter overlay is invoked.)
5. Shared processors written in FORTRAN must be preceded by some Meta-Symbol code that associates the library and links to the FORTRAN code.
6. The root must be greater than one page in length.

When an overlayed shared processor is requested, the processor root and its first overlay are loaded. Assembled data and DCBs are loaded when the root is loaded. Whenever overlays are not required, memory usage can be held down by declaring an overlay length of zero and issuing a CAL to associate that overlay.

Overlays are declared and associated in the same way as they are for batch programs (CP-V/BP Reference Manual, 90 17 64). TREE command cards and M:SEGLD remain the same. CSECT 2 and 3 are converted to CSECT 1 by CP-V loaders.

Shared debuggers (Delta is the only current example) must have only one page of context and no overlays. They reside in the special virtual area of high memory that is currently fixed in virtual (not physical) size in the highest 16K of virtual storage. They may be any physical size less than 16K including their context page.

DATA CONTROL BLOCKS

Most processor I/O operations are performed through standard monitor DCBs. For example, source input is normally read by

M:READ M:SI[options]

The standard DCBs are

M:BI

M:CI

M:EI

M:SI

M:C

M:BO

M:CO

M:DO

M:EO

M:LO

M:SO

M:PO

M:AL

M:LL

M:OC

M:SL

M:GO

The default assignment of monitor DCBs is the operational label of the same name (M:DO is assigned to DO, etc.). The assignment of operational labels to devices is shown in Appendix B. The default assignments for batch operations differ from those of on-line operations. This is done so that a program that writes through LO and reads through SI will automatically use the line printer and card reader for batch operations and the terminal for on-line operations. The logical functions associated with the operational labels are described in the CP-V/BP Reference Manual, 90 17 64.

Details concerning input buffers, error handling, and so on are specified as parameters in a read or write call. Parameters associated with files and devices are specified by the ASSIGN (batch) or SET (on-line) control command.

A processor may construct its own DCBs by means of the M:DCB procedure. However, processors are not required to construct DCBs. DCBs not constructed by a processor will be constructed by the loader. Standard DCBs constructed by the loader occupy 50 words and are connected to a device either by the loader or by an on-line user by means of special terminal commands. The M:DCB procedure must be used if optional parameters such as read or write accounts exceed the allocation of the standard DCBs (Table 34).

DCBs are also provided in library form and may be explicitly called during a load. The sizes of these DCBs are shown in Table 34.

Processors may use nonstandard DCBs, if necessary. Nonstandard DCBs are constructed by the loader if not constructed by the processor. They must be explicitly connected to a device either by an M:OPEN call in the processor or by a SET command issued by an on-line user since no default assignment via operational labels is provided.

It is common practice for a processor to obtain source input through M:SI, to print a source listing through M:LO, and to print diagnostic output through M:DO. However, processor I/O operations are complicated by the fact that an on-line user can connect SI, LO, and DO either to different devices or to the same device (the on-line default assignment for SI, LO, and DO is the terminal). In particular an on-line user may connect two or more of these standard operational labels to the same device. For this reason, processors must take precautions to avoid duplications in printed output. This means that processors must know at all times whether they were called in batch or in on-line mode and what specific device connections have been made for standard DCBs.

Processors may examine DCBs directly to determine when the DCBs are connected to the same device. Fields within a DCB may be referenced relative to the name of the DCB. Fields that may be useful to processors are as follows:

Field	Use
FCD	Bit 10 of word 0 of a DCB. This is the file-closed flag. A 1 means the associated file is open; a 0 means the file is closed.

Table 34. Standard DCBs

Name	Device	Name	Account	Pass-word	Expiration Date	Read Accounts	Write Accounts	INSNS	OUTSNS	Synonymous Name	Key Buffer	Total Words
Loader Built DCBs	22	4	3	3	3	0	0	4	4	0	8	51
M:C	22				3							25
M:OC	22				3							25
M:BI	22	9	3	3	3			4			8	52
M:CI	22	9	3	3	3			4			8	52
M:SI	22	9	3	3	3			4			8	52
M:EI	22	9	3	3	3			4		9	8	61
M:BO	22	9	3	3	3	17	17	4			8	86
M:CO	22	9	3	3	3	17	17	4			8	86
M:SO	22	9	3	3	3	17	17	4			8	86
M:PO	22	9	3	3	3			4			8	52
M:LO	22	9	3	3	3			4			8	52
M:LL	22	9	3	3	3			4			8	52
M:DO	22	9	3	3	3			4			8	52
M:GO	22	9	3	3	3						8	48
M:EO	22	9	3	3	3	17	17	4		9	8	95
M:SL	22	4	3	3	3						8	43
M:AL	22	4	3	3	3						8	43

Field	Use
TYPE	Bits 18-23 of word 1 of a DCB. These bits specify a code for the type of device connected to the DCB (printer, terminal, card reader, etc.).
DEV	Bits 24-31 of word 1 of a DCB. These bits specify an index to the monitor device table.

Under CP-V, all device assignments are direct. This means that DEV always contains a direct device assignment. A complete layout and description of DCBs is contained in the CP-V/BP Reference Manual, 90 17 64.

The same effect can be obtained by the CORRES device CAL, but the CAL is much slower than the direct comparison. The direct comparison of the combined TYPE-DEV fields is meaningful only if the DCB has been opened. This means that processors must explicitly open DCBs for which device assignments will be tested.

FILE IDENTIFICATION

All on-line processors use a common format and common character set for constructing file identifiers (fid). The standard format is

$$\text{name}\begin{bmatrix}\text{.account}\\ \text{.account.password}\\ \text{..password}\end{bmatrix}$$

where name, account, and password consist of character strings with maximum lengths of 11, 8, and 8, respectively (name has a maximum of 31 characters for CCI, Edit, and PCL and a maximum of 10 characters for Link and Load). Any of the following characters may be used:

A-z a-z 0-9 ␣ $ * % : # @ -

Lower case alphabetical characters are not available on all terminals (e.g., Teletype Models 33 and 35). If lower case letters are sent to these terminals, they are printed in upper case.

Account and password are optional. If account is omitted, the log-on account is the default account. If password is omitted, no password is required to access the file.

TEL SCAN

A processor call entered through a terminal via TEL has the form

$$\text{lm [sp]}\left[\begin{matrix}\text{ON}\\ \text{OVER}\end{matrix}\text{ [rom][,list]}\right]$$

where

lm is the name of the processor and is a file identification (fid). Account :SYS is assumed.

sp specifies a source program and may be either a file identification (fid) or a terminal identification (ME).

ON indicates that ROM output is to be on a new file.

OVER indicates that ROM output is to be over an existing file.

rom specifies that the relocatable object module produced by the processor is to be directed to a specified file (fid). If no file is specified, output is directed to a special file that may be subsequently referenced by a dollar sign.

list specifies that a file (fid), a line printer (LP), or the terminal (ME) should be used for listing. If list is not specified, no listing output is produced.

These specifications are implicit ASSIGN and SET commands for the DCBs M:SI, M:GO, and M:LO. A processor call causes the specified processor to be executed with M:SI DCB input from the file sp. Processor output through M:GO DCB is placed in the file specified by "rom" and listing output (M:LO DCB) is directed to the file or device specified by "list". Processor calls are interpreted by TEL.

Parts of a processor call may be enclosed in parentheses. TEL does not do anything to these parts of a processor call. However, the processor may examine these and other parts of the command line that is in its JIT buffer (J:CCBUF).

Processors may reside in storage in three forms:

1. System swap storage contains absolute shared copies of frequently-used processors. These copies can be located and loaded quickly. The absolute shared processor file is created during system initialization and contains reentrant processors that are shared among all concurrent users.

2. The :SYS account may also contain copies of processors in load module form. Processors in this form cannot be loaded as quickly as absolute processors, but the :SYS account may be useful during processor construction, debugging, and extension. Public programs in the :SYS account may be called by entering their names in TEL commands or on control cards.

3. A user may store his own processors or his copies of system processors in his own files (account). A processor stored in a user's file area is identified by its file name and may be called by the RUN command in batch or START command in on-line operations.

When TEL encounters a processor call, it issues an exit CAL specifying the requested processor. The monitor routine STEP searches the shared processor list for the name of the processor. After it finds the name, it determines whether or not a copy of the processor is in core. If the processor is not in core, STEP loads it. If the name of the processor is not in the shared processor list, STEP searches the :SYS account and loads the processor from that file. If the processor cannot be found, an error message is sent to the terminal. Before control passes to the processor, TEL checks the parameters of the processor call for correct syntax and for existence of the "sp" file and a "rom" or "list".

TEL sets and resets bits in JIT to correspond to the commands LIST, DONT LIST, etc., and to the initial occurrence of assignments in the command string. One JIT word (JOPT) contains a bit for each option that can be specified for a processor. The options and their corresponding bit assignments are as follows:

Identifier	Bit	Set	Reset
LO	31	LIST	<u>DONT LIST</u>
BO	30	Unused	Unused
GO	24	<u>OUTPUT</u>	DONT OUTPUT
DO	23	<u>COMMENT</u>	DONT COMMENT

The underlined values are default values. If a SET command is issued for the corresponding DCB, or the list output or binary output fields are specified in a TEL command, the corresponding bits are set. Each processor must assign meaning to the bits and interpret them. Unassigned bits are available for future use. Checks of these bits should be made on each write command since TEL allows on-line users to interrupt the processor and turn on or off the LO, GO, and DO devices.

Each processor should establish conventions to maintain orderly output when two or more DCBs are connected to the same device. The usual convention is that if diagnostic output has been written via M:LO, and M:LO and M:DO are connected to the same device, then the diagnostic output should not be written via M:DO. The following example illustrates some of the special cases that processors should consider:

1. M:SI, M:DO, M:LO connected to the same device (the input line should not appear three times).

2. M:DO connected to a device that is different from SI and LO (the diagnostic comment should probably be printed beneath the line in error).

3. M:SI and M:DO connected to a Teletype (processors may or may not want to type a line in error).

Processors may read each input image via the M:SI DCB. The last record of the sp will cause an end-of-data abnormal condition (see the CP-V/BP Reference Manual, 90 17 04 for a description of abnormal conditions). To obtain control on an error or abnormal condition, a processor must issue the M:SETDCB command and/or include error and abnormal exits in its read and write CALs. Since source input may come from a Teletype (sp = ME), processors must be able to handle Teletype input. The problems associated with Teletype I/O are discussed in the section on terminal I/O.

CCI SCAN

On transferring control to a user's program or to a processor, the monitor communicates the TCB address via general register 0. Processors may fetch the card image of the command that called them by reading through a DCB connected to the C device.

When running in batch mode, the processor must read the C device once to clear the control command. The command is transferred to the user's buffer to allow the user's program to examine parameters.

TERMINAL I/O

An on-line user may direct output to his Teletype at any time during execution of a processor. Similarly, portions of the input to a processor may come from a Teletype. In general, Teletype I/O is the same as other I/O in its use of M:READ and M:WRITE operators and the standard abnormal and error situations. However, Teletype I/O has some features that are significantly different from those for other devices. Some of the differences require special attention by processors, but the interface is designed in such a way that processors will not have to know whether or not I/O operations are via Teletype, providing they observe certain conventions. On terminal I/O, like all I/O, the user should note that byte displacements in the DCB remain in effect until replaced, once they have been given. The special problems associated with Teletype I/O are outlined in the following paragraphs.

END CHARACTERS

On input from a Teletype, each record read is terminated by an end character (CR, FF, LF, RS, US, FS, GS). The end character, if any, is included in the actual record size (ARS) count reported in the DCB (bits 0-14 or word 4). Each processor must interpret the different end characters. Processors do not have to know that input is via Teletype, provided they treat these characters as terminators and use ARS to determine the actual record received.

Source files for all processors, including those in batch operations, may have been prepared on-line. Since records prepared on-line are variable length, it may no longer be assumed that input records are 80-byte card images.

All characters received from terminals, no matter of what type, are translated to the standard EBCDIC character set. The hexadecimal codes for EBCDIC characters are listed in Appendix A.

WRITE OUTPUT

The length of each output line is specified by the SIZE parameter in the M:WRITE procedure call. It is terminated only by the character zero. That is, the user may terminate a message with a zero character if he wishes and the COC routines will compute the proper message length. Carriage return or new line characters do not terminate a message.

CARRIAGE RETURN

A new line or carriage return sequence, as appropriate to the type of terminal, is appended to the character string supplied by each write under the following circumstances:

1. The DCB is not M:UC.

2. The suppress space option is not specified.

Thus, under ordinary circumstances, carriage return characters will be supplied when output consists of one line per write and the DCB is connected to a terminal. By using the suppress space option or by writing through M:UC, the program may supply carriage returns exactly to requirements – either none or several for each write CAL.

PARITY ERRORS AND LOST DATA

When an M:READ CAL specifies a terminal, any character received with a parity error is replaced by SUB (USASCII code 1A) and the lost data abnormal code (07) is returned to the user if an abnormal address exists. If there is no abnormal address, control proceeds to the CAL plus 1. The line is returned to the user's buffer and the program may expect to encounter the SUB code as it scans.

In designing a response to messages that contain parity error characters, two facts are important:

1. The user has already been informed of the error by the COC routines that echo the exact bits received on the line followed by the # character.

2. If the received image is sent back to the terminal together with an error message, the # character will be printed when SUB codes appear.

In the absence of special considerations unique to the processor, it is recommended that lines received with lost data be sent back to the terminal together with the comment "EH?". This procedure is helpful as an aid in diagnosing faulty terminals and communication lines.

END-OF-FILE

If the user types the character pair ESC F, an end-of-file abnormal code will be returned to the program reading the terminal at the abnormal address (if there is one). An input line that contains all characters received prior to the end-of-file sequence will also be transmitted to the user's buffer. This line is always terminated with a carriage return which is also sent to the user's terminal. If no abnormal address is specified, the line appears as an ordinary input line. If both bad data and end-of-file occur in the same input, then the bad data is reported.

OTHER ABNORMAL CONDITIONS

If unknown operations are requested of the COC routines (e.g., write end-of-file), the abnormal code for beginning-of-tape will be returned. If there is no abnormal address, the operation will be ignored.

FORMAT CONTROL

COC routine action for the various formatting CALs is specified in the CP-V/TS Reference Manual, 90 09 07. It is briefly reviewed below.

It is sometimes necessary to print a line with special spacing or without a carriage return. Processors can obtain vertical carriage control by means of two parameters (SPACE and VFC), both of which can be set by the M:DEVICE CAL. The SPACE and VFC parameters have the following interpretations for Teletypes.

Parameter	Meaning
SPACE	If this parameter is set and VFC is not on, the number of spaces indicated minus 1 is inserted before each write. Counts of 0 and 1 result in single spacing.
VFC	If this flag is set, the COC routines simulate the printer's vertical format control as specified in the first character of the text lines written. The simulation is limited to one of the following cases:

Hex. Code	Action
C1-CF	COC inserts 1-15 spaces before printing.
F1	COC skips to top-of-page by skipping six lines and printing the heading information followed by the print line.
60,E0	COC does not insert CRLF after the print line (suppress space).

For page control, COC routines count the number of lines transmitted to and received from the user's terminal. New page headings are printed for every read or write when the line count exceeds the maximum specified in JIT (via the PLATEN command). New page headings are also printed if the user program issues a PAGE device CAL or if the terminal user types the FF character L^c (CONTROL L).

Information in the page heading may be specified by the user by means of the HEADER and COUNT device CALs. Heading information is taken from the DCB through which the read or write was given. Thus, if a write call is issued to a Teletype through more than one DCB, the heading printed depends upon the DCB through which the top line of the page was written. The automatic page heading occupies one line and contains current time, date, user name and account number, user identification and line number, page number, and possibly an administrative message. Headings specified in the DCB of the read or write are produced after the automatic heading with position, text, and page number as specified in the CP-V/BP Reference Manual, 90 17 64. The page count in this heading is that carried in the DCB and is reset with each COUNT device CAL. The page count for the automatic heading is carried in JIT and is never reset. The automatic heading is suppressed if the page length is less than eleven lines. Headings are also not printed if the automatic page heading is turned off via the TEL PLATEN command.

Tab characters are replaced with an appropriate number of blanks in input lines. Tabs are not required in output lines. However, if a highly formatted output line is sent to the Teletype, the operation will be more efficient – and more satisfactory for the on-line user. Tabs are activated by inserting a tab character (X'05') in the output stream. Tabs may be sent directly to the terminal or simulated by the software as requested by the terminal user who may turn simulation on and off using the sequence ESC T. When simulated by the software, each tab character in the output stream causes insertion of spaces to move the carrier to the right to the next higher position specified in the DCB.

Simulated tab stops can be set by a processor with the TAB device CAL or by an on-line user (for the M:UC DCB) with the TABS command. Tabs must be specified in ascending order beginning with tab stop position 1. Note that this is different from the line printer tabbing, where the tabs need not be in ascending sequence. Tab stops can be set at any time for any DCB. During output operations, tabs are expanded as specified by the DCB through which the write is issued or, if not specified there, as specified in the M:UC DCB. Tabs typed by an on-line user are simulated at the user's console according to the tab settings in the M:UC DCB.

If the backspace character is typed at the terminal, the character is passed to the reading program. No special action is taken by the COC routines other than that necessary to record current carrier position (which for backspace depends on terminal type). Terminals that have a physical backspace may, at the user's option, use a "backspace-edit" mode for intra-line editing. (Reference: CP-V/TS Reference Manual, 90 09 07.)

A program can request control when the user presses the BREAK key by means of the M:INT procedure. Whenever the user presses the BREAK key, the program environment at the time of the break is recorded in the user's pushdown stack in his TCB. Execution can be returned to the location following the interrupted instruction by execution of the M:TRTN procedure. A program can return break control to TEL by executing the M:INT procedure with a break routine address of zero. The break routine address is checked by the monitor to guarantee that the address lies within the memory allocated to the user. Even if a processor has obtained break control, an on-line user can return execution control to TEL by pressing the ⓔⓢⓒ ⓔⓢⓒ, ⓔⓢⓒ Y, or Y^c keys.

As a safety measure to protect the user against faulty programming in break control routines, the number of times the BREAK key is pressed by a user without intervening characters is recorded. When the count reaches four, control is sent to TEL as if Y^c had been pressed. Thus, the user at the terminal will never find himself locked out. The count of four allows processors (e.g., FDP) to make special interpretations on two and three breaks in a row.

FILE EXTENSION

File extension is a convention by which records are added to an output file by successive job steps. Each time the file is opened, the file pointer (tape, disk pack, etc.) is positioned to a point immediately following the last record in the file. Thus, when additional output is produced it is added to the previous contents of the file, thereby extending it. File extension simulates output to physical devices, such as line printers or typewriters, when output is actually directed to a file.

File extension takes effect at the time CP-V opens system output DCBs. The output DCBs that are affected by file extension are those that are currently assigned to files, although normally assigned to devices. They include: M:LO, LL, DO, PO, BO, SL, SO, CO, AL, EO, and GO.

File extension is discontinued when a file is reassigned with a SET or ASSIGN command or when a file is opened with an OPEN procedure call that specifies an explicit file name. In these cases, a new file is created. Extension of the GO file is terminated following a LINK or RUN command.

SHARED FILE USE

Shared processors must ensure that temporary files used during operation are distinct for each instance of execution. A common technique for accomplishing this is to append the current users ID, from the right half of the first word of JIT, to the filename when it is created and used. This ID is guaranteed by the system to be unique for all concurrently running batch or on-line programs. A discussion of shared files is contained in CP-V/BP Reference Manual, 90 17 64.

COMMAND PROCESSOR PROGRAMMING

A command processor is a shared processor which interfaces between the user and that which the user wants to access — the monitor, a processor, or another program. Four command processors are supplied with CP-V. They are LOGON, TEL, CCI, and EASY. CP-V will also support installation-specific command processors. Information about the programming of command processors is outlined below.

Generally, command processors have the same restrictions as listed for shared processors previously. In addition:

1. A command processor may not have any overlay structure.
2. A command processor which resides in the special processor area (above X'1C000') may not have any dynamic data and must be biased at X'1C200'.
3. A command processor must intercept all exits, errors, and aborts from user programs and must clean up correctly. (Special CALs for command processors are listed below.)
4. Command processors should not be given special JIT access. (The special CALs for command processor interface eliminate the need for it.)
5. When programs error or abort, control will be given to the command processor with the following restrictions:

   If the command processor resides in the user program area (X'C000' to X'1C000'), the exiting user program will be completely disassociated before associating the command processor, eliminating the possibility of continuation of the job step.

   If the command processor resides in the special area (X'1C200' to X'1FFFF'), has no dynamic data or DCBs, uses only M:UC, M:OC, and M:XX, control will pass to the command processor with the user intact, allowing analysis of the exit and continuation of the current job step.

Command processors may be entered into the system during PASS2 of SYSGEN by using the T, B, G, and C flags of the :SPROCS command. They may also be added to the system, replaced, or deleted from the system via the DRSP processor.

The following capabilities are available to command processors:

1. Interpretive Exit – An interpretive exit is a normal exit CAL (M:EXIT) performed by a command processor with the following register setup required.

R6, R7, R8	Contain the TEXTC name of the requested load module or shared processor. A maximum of seven bytes is allowed for a shared processor. If R6 is zero and the command processor is special shared, (biased at X'1C200'), the program is reentered at the point of interruption.
R13, R14	Contain the account (in TEXT format) in which the load module resides. :SYS is specified for shared processors.
R10, R11	Contain the password in TEXT format. If there is no password, zero should be used.
R0, R1	Contain either FDP or DELTA in TEXTC format or a zero. If one of the two debuggers is specified, the interpretive exit is to be taken with the debugger associated.

The system job step processor, STEP, interprets such an exit as a call on the specified program. It also loads the TEXTC name of the command processor that issued the interpretive exit into R4 and R5. Before a command processor issues an interpretive exit, it must have closed all its DCBs and, in general, have cleaned up. The job step processor arbitrarily removes the command processor from the user's virtual map. This means that all data and DCBs are gone.

2. BREAK and CONTROL Y Control – If the terminal user depresses the BREAK key during operation of a processor or user program and that program did not request BREAK control, the program is aborted and the command processor is loaded and entered with bit 30 of J:TELFLGS in the JIT set. If the interrupted program has requested BREAK control, the program's BREAK routine is entered.

If the terminal user depresses CONTROL Y during the execution of a processor or user program and the command processor is not special shared, the program is aborted and the command processor is loaded and entered. If the command processor is special shared and has no data and no DCBs, the user program is left as is and the command processor is entered. This gives the command processor the opportunity to continue the interrupted program.

If the terminal user depresses CONTROL Y while a command processor is in control, the event is ignored and the current operation is continued where it was interrupted.

If the terminal user depresses BREAK while a command processor is in control and BREAK control has not been requested or BREAK control has been reset via the M:INT CAL, the BREAK event is ignored and the command processor is continued where it was interrupted. If a command processor has requested BREAK control, it is interrupted at its BREAK control address.

The format of the BREAK control CAL is:

```
CAL1,8    FPT
```

where FPT points to word 0 of the FPT shown below.

Word 0

X'00'	0———0	BREAK routine address
0 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

3. Exit, Error, Abort CAL, and I/O Abort Control – If any exit or abort condition occurs during execution of a program, the program is aborted and the command processor is loaded and entered. Error conditions are described in four fields of the JIT as follows:

- J:ABC is the address of the word in the JIT that contains the abort code in byte 0 (see Appendix B of the CP-V/TS Reference Manual, 90 09 07).

- ERO is the word offset into the JIT of the word that contains the abort subcode in byte 3.

- J:RNST contains the address of the word in the JIT that contains the current run status. Status settings are:

All zeros	means the job is executing normally.
Bit 1	if set, the job is to be errored because of an M:ERR call to the monitor.
Bit 2	if set, the job is to be aborted because of an M:XXX call to the monitor.
Bit 3	if set, the job is to be errored because of an E key-in by the operator.
Bit 4	if set, the job is to be aborted because of an X key-in or a line disconnect.
Bit 5	is reserved for future use.
Bit 6	if set, the job is to be aborted because a limit has been exceeded (e.g., maximum pages out).
Bit 7	if set, the job is to be aborted because of an error (most likely I/O) as specified in J:ABC and ERO.
Bit 8	if set, the job is to be aborted because of an illegal trap.

- J:ASSIGN contains the address of the word in the JIT, the rightmost nine bits of which indicate which limit was exceeded. This field is set in conjunction with bit 6 in the RNST field of the JIT. The bits, if set, mean:

Bit 23	the maximum disk allocation limit exceeded.
Bit 24	the maximum time limit exceeded.
Bit 25	the maximum scratch tape limit exceeded.
Bit 26	the maximum temporary disk space limit exceeded.
Bit 27	the maximum permanent disk space limit exceeded.
Bit 28	the maximum diagnostic pages output limit exceeded.
Bit 29	the maximum user pages output limit exceeded.
Bit 30	the maximum processor pages output limit exceeded.
Bit 31	the maximum punch output limit exceeded.

4. CAL Control of JIT Error Condition – This CAL allows control of JIT error conditions without special JIT access. The form of the CAL is:

   CAL1,4 fpt

   where fpt points to the word shown below.

X'06' (bits 0–7)	0 ———— 0 (bits 8–31)

   The monitor (the ALTCP portion) verifies that the program issuing the request is a command processor through use of UH:FLG. It then sets J:ABC, ERO, byte 0 of J:RNST, and bit 30 of J:TELFLGS to zero. (Bit 30 of J:TELFLGS indicates whether or not the BREAK key has been depressed.) If the program issuing the CAL is not a command processor, control is returned to the user program with CC1 set.

5. Registers – Upon entry to a shared processor from a command processor, the registers must contain the following:

R0	the TCB address of the user program.
R4, R5	the name of the calling command processor in TEXTC format.
R6, R7, R8	the name of the called processor in TEXTC format.
R10, R11	the password in TEXT format (zero if none).
R13, R14	the account of the called processor in TEXT format.

6. CAL Control of Terminal Modes – Control of terminal modes is provided by a variation of the Change Terminal Type CAL (see the CP-V/BP Reference Manual, 90 17 64).

PUBLIC LIBRARIES

The system may have several shared public libraries. Each library is a unit tailored to the requirements of the installation. The user associates a public library with his program by specifying the library name (Pi where i = 0 – 9 or J0) in a LINK or RUN command. The rules governing library units are as follows:

1. Link loads the user data immediately above the area reserved for the library data. Load reserves an entire page for library data.
2. No initialization is provided for this temporary library data either by the loader or by the system. There must be an initialization program if initialization is required.
3. Each library unit must separate data (CSECT0) and program (CSECT1) information into separate assemblies so that separate ROMs will be produced for each.
4. All code must be under CSECTs with protection type 0 for variable data or 1 for procedure and constant data. No DSECT section may be used.
5. The library must be self-contained (i.e., there can be no unsatisfied references). This must be true for the data portion itself and the total library. For example, a FORTRAN I/O library must search the DCB chain rather than make a direct reference to the DCB itself.

CP-V PUBLIC LIBRARIES

CP-V contains three public libraries. One library (:P1) includes the most commonly required routines from the Extended FORTRAN IV and Extended FORTRAN IV-H library (about 65 routines). Another (:P0) includes :P1 plus the FORTRAN Debug Package (FDP). The third library (J0) contains the JIT definition. Most executing users need only the first library; users who are debugging need the second.

The entire Extended FORTRAN IV and Extended FORTRAN IV-H library consists of 266 routines (ROMs) totaling more than nine thousand instructions and over 800 data words.

The package includes more than 350 DEFs. These routines are described in Extended FORTRAN IV Library Technical Manual, 90 15 24 and Sigma 5/7 Mathematical Routines Technical Manual, 90 09 06.

Public library :P1 contains single and double precision trigonometric functions, exponential and logarithmic functions, standard set-up routines, initialization and termination routines, and input/output conversion and transmission routines. Fewer than 300 words of storage are required for temporary storage by each user of the library. The 4800 words of library code are shared among all concurrent users.

FDP users require public library :P0 which consists of nearly 800 words of temporary storage per user; over 9700 words of code are shared among the concurrent users.

The remaining 170 routines of the complete FORTRAN library are organized in two ways:

1. They are organized in the :BLIB file as card-image ROM decks that are used by the Link loader to satisfy library references.

2. They are organized in the :LIB/:DIC files as 19 library load modules.

This organization permits rapid loading by the batch loader (Load). Load uses the file :DIC, which consists of a record keyed by each DEF in :LIB and the group number as its value to find the LM names necessary to satisfy references.

One essential monitor subroutine must be added to the standard released library, S:OVERL. It is normally added during the System Generation process but must be remembered whenever a new library is being installed.

The sizes and description of routines in :LIB are given in Table 35.

CREATING PUBLIC LIBRARIES

User's may add their own public libraries to meet specific requirements. The necessary procedures are given below.

The procedure for creating public libraries consists of several steps. The desired data and program elements are loaded, and the dictionary for the library (DEFs) is filed for loader use. Next, the procedure is filed so that SYSMAK can place it on swap storage during system initialization. In the process, the program SYMCON is used to retain only those DEFs required in the final linking process, thus saving loader stack search time. Figure 36 illustrates the process of creating a public library.

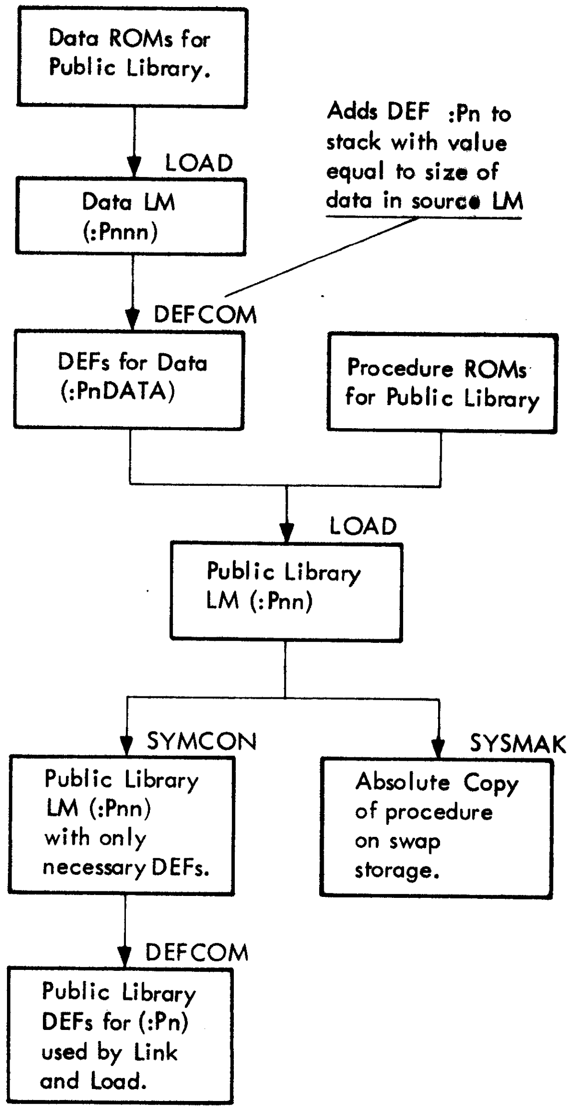
LOADING PUBLIC LIBRARIES

Default loading for Link includes the basic FORTRAN public library (:P1) and a search of the system (ROM) library if there are unsatisfied references. This is the same as if the user had specified (:P1) in a RUN or LINK command. If the user has not explicitly asked for :P1 and no reference to 9INITIAL is found, the procedure for :P1 is not associated with the user program execution although the 350 data words remain committed because of the single pass loader operation. Figure 37 is a generalized flow of the Link process relative to libraries.

Since the batch loader operates in two passes, it makes an explicit association of :P0 and :P1 to a program in absence of other instructions. This process is illustrated in Figure 38.

Table 35. Routines in :LIB Library File

Group	Size	Description
1	96	Complex double precision mathematical routine drivers.
2	72	Complex mathematical routine drivers.
3	108	Double precision mathematical routine drivers.
4	192	Single precision mathematical routine drivers.
5	260	External revisions of compiler intrinsic functions.
6	676	Complex double precision mathematical routines.
7	514	Complex single precision mathematical routines.
8	114	Double precision mathematical routines.
9	112	Miscellaneous integer functions.
10	144	Miscellaneous real functions.
11	98	Logical functions.
12	30	Conversion routines.
13	326	DSINH, DTANH, DASIN, DTAN.
14	230	FORTRAN II special functions.
15	24	Overflow and divide check.
16	256	Buffer-in/buffer-out.
17	668	Input and INPUTL.
18	114	Random Disk I/O.
19	514	Disk buffer.



Note: n = 0–9 for public libraries 0–9.

Figure 36. Public Library Creation Process

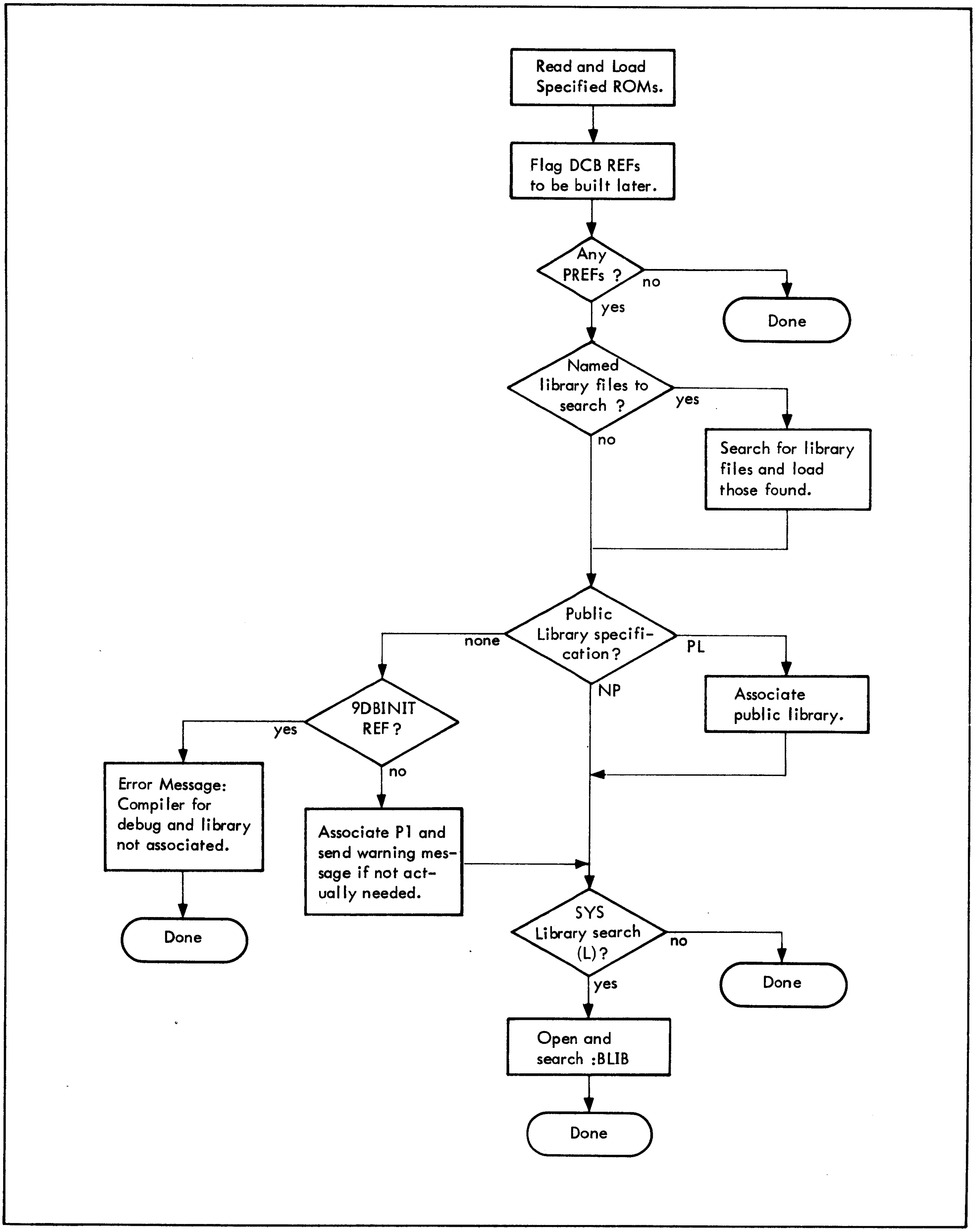
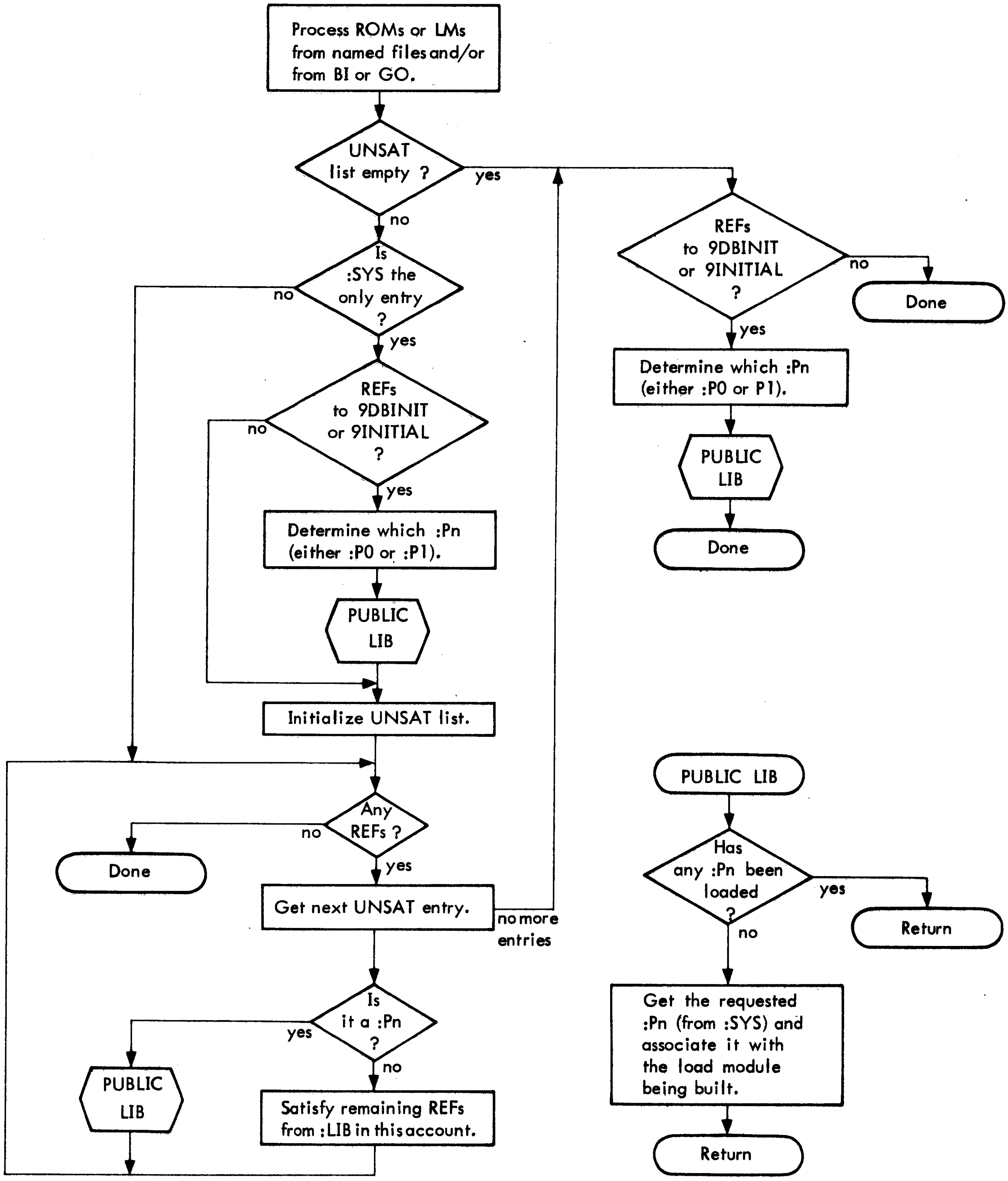


Figure 37. Generalized Library Load Process (Link)



Note: If NOSYSLIB is present on the !LOAD card, the UNSAT list is empty or consists of those sources (accounts and/or :Pn) mentioned under the UNSAT option. If NOSYSLIB is not present, the UNSAT list consists of the above plus the :SYS account (which occurs last).

Figure 38. Generalized Library Load Process (Load)

8. MAINTENANCE OF THE FILE SYSTEM

OVERVIEW

CP-V provides a variety of processors designed to meet the need for maintaining a reliable backup of the file system. A reliable backup of the file system is required for several reasons.

- The hardware may fail, either resulting in physical damage to the storage device or, more likely, presenting bad data to the software which may cause loss of or damage to the files.
- In the course of operation, demand for file space often exceeds availability, in which case it is necessary to move older files to secondary storage to make room for newer files.
- Privileged users (C0 or greater) can potentially damage the file system.
- Scheduled maintenance of the hardware sometimes requires use of the file devices by the Customer Engineer.
- Users make mistakes and delete files that they really wanted to keep.

FILE MAINTENANCE PROCESSORS

The following paragraphs provide brief descriptions of the file maintenance processors. Each processor has a function for which it is uniquely suited; together, the processors comprise a flexible mechanism designed to meet most file maintenance needs. These processors are described in detail in the CPV/OPS Reference Manual, 90 16 75.

FSAVE

The FSAVE processor is designed to save the bulk of the file system in the shortest time. It saves files at or near device speed, and it interfaces with another processor, GAC (described later in this section), to provide correct data for day-to-day file usage. The price paid for speed is significant system degradation; other users will receive little or no time while FSAVE is running. FSAVE does not detect that some files have not been modified since they were last saved and, therefore, some files are saved redundantly. FSAVE does not flag files as having been saved nor does it remember the latest content of each file directory.

FRES

The FRES processor is designed to restore a large volume of files in the shortest time. The files are not restored at device speed, however, since in addition to restoring files, FRES marks the files as having been backed up. This is important for efficient operation of the FILL processor, described below. Note that an existing file will be overwritten by the incoming file without a date verification which may not always be desirable. A further limitation to be noted is that FRES requires files to be in alphabetical order (by name and account) on the tape. While tapes for FSAVE, FRES, and FILL are generally compatible, the files on a SQUIRREL tape created by FILL may possibly be out of sequence so that the tape will not be restored properly when FRES is used.

FILL

The FILL processor performs all of the functions of FSAVE and FRES, but at a slower speed. Additionally:

- It recognizes the files that have already been saved and avoids duplication of effort.
- It allows user initiation of file backup via the !BACKUP command.
- It ensures that each file being restored is newer than the one already existing.
- It does not automatically restore files that users have deleted.
- It provides a mechanism (PURGE) by which files are systematically removed to secondary storage.

Note that all of the file maintenance processors fail to update the :USERS file to account for restored or purged files.

HGPRECON

The HGPRECON processor rebuilds the allocation tables used by file management in the building of files. The allocation tables are correct following HGP reconstruction, but individual files or accounts may have been damaged or deleted. This is due to the fact that dual allocation is avoided by truncating the second file requesting the granule. Files may also be damaged when bad links are detected. Such files are also truncated at the last good link. This is done because dual allocation poses a greater danger to the file system than lost data to an individual file. No data is lost, however, if a FILL SQUIRREL is run just prior to booting from disk to do the reconstruct. In this case, all files that could possibly be damaged by HGPRECON that had retrievable data will be recovered.

PACKRECON

The PACKRECON processor performs for a private volume set the same service HGPRECON performs for the public file system. The private volume allocation tables are rebuilt in the same manner that ALLOCAT's tables are rebuilt. PACKRECON requires that all volumes be in good enough condition to be mounted by the system and they all be mounted simultaneously.

GAC

When files are restored and deleted by file maintenance processors, the individual accounts are not properly credited. The necessary correction of the accounts is the purpose of the GAC processor. GAC uses the output generated by FSAVE (created using the +LOG and +STATS options) to update the :USERS record for each user account to reflect the number of granules currently in use in that account. It also flags accounts which contain more granules than are authorized as well as those accounts which are not authorized at all.

DEVDMP

The DEVDMP processor produces a device copy of the RAD or pack. That is, individual sectors are copied to tape in the order that they actually occur. DEVDMP is not recommended as a file saver because "files" do not exist in this context. A device dump, however, is useful in the following circumstances:

1. If the swapping RAD is destroyed, a previously taken dump provides a copy of the swap device that can be restored. The dump includes the allocation tables and the resident system. However, beware that if the swapping device includes any file space, restoring the swapper from a DEVDMP tape can seriously damage the file data base. HGPRECON can correct the data base to be internally consistent but files and data may be lost.

2. Preventive maintenance sometimes requires the use of a device which contains file data. Running DEVDMP prior to turning the system over to Customer Engineers permits bringing up the system without doing a full file restore.

RECOMMENDED USAGE

The following guidelines are presented for each of the indicated functions and should be modified according to the needs of the individual installation.

SAVING FILES

The entire file system should be saved periodically via FSAVE with the LOG STATS option included. This provides a snapshot of the system and allows any granule accounting errors to be corrected by GAC. If a full restore is required, this snapshot provides the checkpoint beyond which no other tapes need be restored. Note that if an FSAVE SAVEALL is performed on a new system which has not been using FILL, it should be followed immediately by an FRES ALL. This allows FRES to update the file system backup dates so that FILL will function properly.

Following the FSAVE, a schedule of FILL INCREMENTAL saves should be established (e.g., each day during light system load periods). If system data is critical or the system is experiencing difficulties, SQUIRREL saves should be scheduled between the INCREMENTALs.

In order to protect the file system and provide for complete restoration in case of disaster, no tapes prior to the SAVEALL need by saved. However, it has been found that older tapes should be saved because

1. Tape errors can sometimes be resolved by using prior tapes.

2. Users may want to recover files which they have deleted and which only exist on older tapes.

Example:

1. Boot a new system. Restore any previous file base from any available source.

2. Checkpoint:

   FSAVE ALL LOG STATS
   FSAVE DISKPOOL
   FRES ALL (including DISKPOOL)
   GAC

3. Schedule seven days of INCREMENTALs.

4. After the INCREMENTALs on the seventh day:

   FSAVE ALL LOG STATS
   GAC

5. Repeat steps 3 and 4.

To protect against total system failure, a DEVDMP of the system swapping device is recommended at least once after system startup. Although the need for such a backup is remote, it could save valuable files if available.

RESTORING FILES

Proposed here are two methods to restore the entire file system. Depending on the particular characteristics of the files restored, one method may be faster than the other. The end result is the same.

Method 1:

1. Boot from the PO tape. This is better than booting under the files. On a fresh boot, it is not necessary to delete files that may already exist. This speeds the restore process considerably.

2. Use FRES to restore the most recent SAVEALL set of tapes.

3. Use FRES to restore FILL INCREMENTAL tapes in the order created. Do not restore the last INCREMENTAL set created.

4. Restore SQUIRREL tape sets created since the last INCREMENTAL with FILL. These should be restored in reverse chronological order, i.e., the newest first.

5. Restore the last INCREMENTAL using FILL. This will delete all files which the user had deleted, preventing the restoration of files the user had deleted.

6. Run FSAVE with the LOG and STATS options followed by GAC. This will ensure that each user account is properly credited.

Method 2:

1. Boot from the PO tape as in Method 1.

2. Use FRES to restore the most recent SAVEALL set of tapes as in Method 1.

3. Use FILL to restore SQUIRREL tapes created since the last INCREMENTAL. These should be restored in reverse chronological order.

4. Using FILL, restore the INCREMENTAL tapes next, starting with the second most recent and working back to the last SAVEALL.

5. Restore the last INCREMENTAL using FILL as in Method 1.

6. Run FSAVE with the LOG and STATS options followed by GAC as in Method 1.

Restoring an individual user's files is the job of selective FILL. It will restore any file created by any of the processors (FSAVE, FILL) in a timely manner. Also, it checks to see that the file being restored is newer than any which currently exists. Restoring an entire account presents a special problem, as the files may be scattered among several sets of backup tapes. The following procedure is the most efficient method of restoring an entire account using selective FILL:

1. Restore SQUIRREL tapes containing the account's files which were created since the last INCREMENTAL (in reverse chronological order).

2. Restore INCREMENTAL tapes containing the account's files in reverse chronological order. Start with the most recent set and work back to the last SAVEALL.

3. Restore files from the SAVEALL last. This procedure will put the account back to the point when the last backup was performed. Deleted files may have been restored as this check is not made for selective FILL.

RECONSTRUCTING PRIVATE VOLUME

The PACKRECON processor is designed to correct two problems. First, cylinders may be lost to the volume set when the system crashes. Second, files on the pack may be inaccessible due to some errors. PACKRECON will correct these problems with the following reservation: Files are not "final". PACKRECON only ensures that there is no dual allocation and that the file management tables are internally consistent. Files or parts of files may be lost due to truncation when an inconsistency is detected.

GRANULE ACCOUNTING

To provide the most accurate granule accounting information, GAC should be run after every restore or purge of the file system. Practically speaking, this may not be feasible. Single file restores are the most common occurrence, and they generally have negligible impact on the overall granule usage. An entire account, however, can easily be taken care of using the partial option of GAC. GAC should always be run after a full restore of the file system. An up-to-date :USERS file will prevent users from abusing the file system by using more granules than they are authorized to use.

SAVING A DEVICE

The DEVDMP program should be used in conjunction with scheduled preventive maintenance. It should also be used as the last line of defense against lost files resulting from system failure. At least one copy of the system swapping device should be made using DEVDMP (for use in future emergencies). DEVDMP should also be used to save the contents of a device scheduled for preventive maintenance. This will permit the system to be brought up quickly without having to restore the entire file system after the device is returned.

EXCEPTIONAL PROCEDURES

The final section of this chapter deals with procedures to follow when problems occur which cannot be handled by the above procedures. These include solid hardware errors, dual allocation, and other forms of file damage.

MINOR PROBLEMS

A problem is minor, for the purposes of this discussion, if it affects individual users on a random basis. Random scattered device errors is an example which could lead to damaged files.

The best cure for this type of problem (indeed, the best cure for all file problems) is to quiesce and bring down the system, run a SQUIRREL to save all modified files, and then do a full restore of the file system. This, however, can be very time consuming and the problem may not warrant the time spent as the best solution. A quicker method is to bring the system down and run a SQUIRREL as above, then ZAP the system, boot from disk, and answer Y to the HGPRECON question. When reconstruction is complete and the system is up again, restore the users files as necessary.

MAJOR PROBLEMS

An example of a major problem is a solid hardware error. The occurrence of an I/O ERR 7504 is another indication of major problems. The best procedure is again to quiesce and bring down the system and run a SQUIRREL. This may not be possible, in which case files created or updated since the last backup will be lost. The next step is to correct the hardware problem. When corrected, the file system should be completely restored from backup tapes.

A quicker solution is provided here with the understanding that it may fail. This procedure is:

1. Quiesce and bring down the system if possible.

2. Run a SQUIRREL if possible.

3. Use DEVDMP to copy the system swap device to tape, as well as any of the RADs which may be affected. Pull the packs and turn the problem over to Customer Engineering.

4. When the Customer Engineers have finished, restore the DEVDMP tapes and replace the packs.

5. Boot from the swap device and perform HGP reconstruction.

6. Restore lost files as they are needed.

CATASTROPHIC PROBLEMS

A damaged swapping RAD and a software check 1A are examples of disasters. Again, some files may be lost but the following procedure will get back as many files as possible, perhaps all of them.

1. Restore the latest DEVDMP tape of the system swapper.

2. Boot from the swapping device and perform HGP reconstruction. THIS IS CRITICAL! The DEVDMP tape contains ALLOCAT's data from a previous time. Any attempt at booting without reconstructing this data will cause loss of data and probably the entire file system.

3. When reconstruction is finished, perform an instant SQUIRREL. This will save all files that can possibly be saved.

4. Restore the entire file system.

The last step is recommended, although it may not be necessary. Depending on errors reported by HGPRECON the system manager should choose whether or not to proceed with normal processing.

9. MONITOR DUMP ANALYSIS PROGRAM

INTRODUCTION

The monitor dump analysis program (Analyze) is designed to aid in the debugging of CP-V crash dumps. Analyze operates in the ghost, on-line, and batch modes. It accepts as input any tape or disk dump produced by the recovery procedure and any tape dump produced by executive Delta. If a tape is input, the Analyze user must supply the tape type in response to the message

```
ENTER TAPE TYPE: 9T, BT, ETC...
```

Tape input results in the creation of a disk file (CP5DUMP); subsequent tape inputs replace the contents of this file.

GHOST MODE

Analyze is called automatically by the recovery procedure, and functions as a ghost job to interpret and summarize critical monitor tables and to dump the monitor's dynamic data area. When started in this manner, Analyze neither looks for nor accepts any commands, operating entirely on default options. It assumes an INPUT command option of LAST; if unable to open the last MONDMP file, it then assumes an INPUT command of TAPE. (Refer to the description of the INPUT command in the following text.) When Analyze is run as an automatically called ghost job, the output is an abbreviated form of the output produced by the ALL display command.

Analyze may also be called as a ghost job by the operator to examine the tape produced during an irrecoverable crash. The operator key-in used for this purpose is

GJOB ANLZ

Analyze then asks the operator for a command:

```
ANLZ: ENTER COMMAND, N/L SAYS TO DO ALL
```

The operator may respond with one of the following commands:

NO – just exit.

TA – read a recovery-built tape.

HE – run interactively from the operator's console.

CP – read the CP5DUMP file.

0-7 – read the indicated MONDMP file.

NL (new line alone) – do default ghost run.

In the interactive ghost mode, a key-in of

INT id

will cause termination of the current Analyze operation and a prompt for input. (id specifies the Analyze user's number.)

BATCH AND ON-LINE MODES

Any batch or on-line user may call Analyze by specifying the name of the program (ANLZ). For on-line users, this program name is entered in response to a TEL prompt for commands, as follows:

!ANLZ

Any user, in batch or in on-line mode, must have the proper privilege level (80 or better) to examine the monitor. If he does not, Analyze outputs the following message

```
xx PRIVILEGE LEVEL NOT HIGH ENOUGH
```

where xx is the user's current privilegel level. (Response messages are output on the line printer for a batch user.)

When accessed on-line, as an interactive ghost, or as a batch job, Analyze is completely command-driven. It responds to commands that selectively display monitor tables, examine memory, and compare the dump with the running monitor.

An on-line user may terminate a display by depressing the BREAK key.

COMMANDS

When Analyze is first entered, it responds

```
ANALYZE HERE
```

and, if in on-line mode, it requests entry of an input command with the prompt character

<

All commands, options, and output are identical for batch, interactive ghost, and on-line modes.

INPUT COMMAND

INPUT The INPUT command directs Analyze to input from a particular disk or tape file, or to open a file. The format of the command is

IN[PUT] option

where option may be any one of the options shown in Table 36.

After reading a tape or disk file as directed by the INPUT command, Analyze informs the user of the size of the file with the following message:

THE LAST PHYSICAL PAGE IN THE FILE IS xx

If in on-line mode, it then prompts (<) for the next command.

Table 36. INPUT Command Options

Option	Meaning
T[APE]	Directs Analyze to read a tape created by the recovery process and to write it into the file CP5DUMP which is then used for input.
C[P5DUMP]	Directs Analyze to open the CP5DUMP for input.
L[AST]	Directs Analyze to open the last file formed by the recovery procedure for input. (Analyze must look at the running monitor to obtain this information.)
number	Directs Analyze to open a crash file formed by recovery. Recovery file names are of the form MONDMP(number) where number is the number of the dump file (1 for the first dump since a "cold" start, 2 for the second, and so on).

DISPLAY COMMANDS

Three display commands may be used to output information from crash dumps. They are

DISPLAY

RUN

ALL

DISPLAY The DISPLAY command outputs information existing at the time of the crash. The format of the command is

DI[SPLAY] option

where option specifies the information to be displayed (Table 37).

RUN The RUN command outputs various linked lists of the monitor by running through the list and displaying each entry. The format of the RUN command is

RU[N] option

where option specifies the list to be printed (Table 38).

ALL The ALL command performs all of the functions of the display commands described above and the functions of Analyze (except dumps) when it is initiated by the Automatic Recovery Procedure. The format of the command is

AL[L]

A numerically and alphanumerically sorted monitor map is output at the end of the ALL display.

INTERACTIVE MONITOR DISPLAY COMMANDS

Commands in this group allow the user to examine either the dump or the running monitor. Both the monitor and user JIT and physical core may be examined. The commands are

loc

loc_1, loc_2

Line feed (or carriage return)

↑

*

MONITOR

loc = value

loc The loc command outputs the contents of the specified location. The format of the command is

loc

where loc is a hexadecimal number or an expression indicating a sum or difference of two hexadecimal numbers. Note that loc values do not require a preceding delimiter character ".".

Table 37. DISPLAY Command Options

Option	Meaning
AJ[ITS]	Displays JIT, AJIT, and context area of all incore users.
AT[ABLE]	Displays the items in ALLYCAT's tables used to manipulate buffers (Figure 50).
AV[R]	Displays the AVR tables (Figure 60).
CI[TS]	Displays the CITs (Figure 54).
CO[C] $\left[, \begin{Bmatrix} S \\ id \end{Bmatrix}\right]$	Displays the contents of COC tables (Figure 58). S specifies all users and id specifies a specific user. The default is S.
CU[N]	Displays the current user's JIT, AJIT, and context (Figures 65, 66, and 67).
DC[TS]	Displays the DCTs (Figure 56).
EL[OG]	Displays the incore error log buffer (Figure 61).
FQ	Displays I/O tables not currently in use.
ID $\left[, \begin{Bmatrix} S \\ id \end{Bmatrix}\right]$	Displays user's logon identification as it appears in his JIT. S specifies all users and id specifies a specific user. The default is S.
IO[Q]	Displays channel information (CIT), device control (DCT), and I/O queue (IOQ) tables (Figures 55, 56, and 57).
IQ	Displays the IOQs (Figure 57).
JI[T]id[, loc_1, loc_2]	Displays the contents of JIT (between displacement locations loc_1 and loc_2) for a particular user (Figure 65) where id specifies the user identification assigned by the system.
MR	Displays the monitor's root (Figure 70).
OJ[ITS]	Displays the JIT of all users not in core.
PA[RTITION] $\left[, \begin{Bmatrix} S \\ id \end{Bmatrix}\right]$	Displays the partition table values (Figure 38). S specifies all users and id specifies a specific user. S is the default.
PF[ILE]	Displays the patch file created at system boot time.
PM	Displays the contents of the page matrix, which identifies owners of physical pages.
PP, pgno	Displays the contents of a physical page of memory.
RA[T]	Displays the resource allocation tables (Figure 59).

Table 37. DISPLAY Command Options (cont.)

Option	Meaning
RE[GISTERS]	Displays the contents of the registers and the explicit cause of the crash (Figure 39).
RC[XT]	Displays the recovery context.
ST	Displays the output symbiont tables (Figure 62).
SW[AP]	Displays the contents of the swap tables (Figure 44).
SY[MBIONTS]	Displays all input symbiont table values (Figure 62).
TR[APS]	Displays the contents of trap and interrupt locations (Figure 40).
TS[TACK][,[id][,#levels]]	Dumps the indicated number of Temp Stack levels (default 21_{10}) for the indicated user (default id = 0, which indicates the monitor's stack), displaying values as symbol plus displacement. If the stack cell contains a monitor address, the instruction at that location will be displayed if it is an address-modifying instruction (e.g., BCS, BAL, or XPSD). (Figure 66.)
US[ER]$\left[,\begin{Bmatrix}\text{S}\\\text{id}\end{Bmatrix}\right]$	Displays the contents of the tables for a particular user (Figure 42). S specifies all users and id specifies a specific user. S is the default.
VP, pgno	Displays the contents of a virtual page of memory. Note: To make a user's virtual memory available for this display, the MAP command must first be entered.

Table 38. RUN Command Options

Option	Meaning
PR[OCESSOR]$\left[,\begin{Bmatrix}\text{S}\\\text{name}\end{Bmatrix}\right]$	Specifies processor pages or specific processor (Figure 46). The default is S, indicating all processor pages.
US[ER]$\left[,\begin{Bmatrix}\text{S}\\\text{id}\end{Bmatrix}\right]$	Specifies user pages for all users (S), or for a particular user (Figure 42). The default is S.
MO[NITOR]	Specifies monitor pages (Figure 47).
ST[ATE]$\left[,\begin{Bmatrix}\text{S}\\\text{q\#}\end{Bmatrix}\right]$	Specifies state queues (Figure 43). The number of a specific state queue may be specified (q#), or S indicates all. The default is S.

loc_1,loc_2 This command outputs the contents of the memory locations between loc_1 and loc_2. The format of the command is

loc_1, loc_2

where loc_i is a hexadecimal number or an expression indicating a sum or difference of two hexadecimal numbers.

Two levels of loc_i commands may be joined by + and - operators. For example, the following are permissible:

loc + loc_1, loc_2

loc - loc_1, loc_2

loc_1, loc_2 + loc

loc_1, loc_2 - loc

loc_1 + loc_2, loc_3 - loc_4

The resultant dump suppresses identical lines and an * is inserted next to the line number following the identical line encountered. An EBCDIC translation is included to the right of the dump.

LINE FEED The line feed (or carriage return) character may be used in conjunction with loc and loc_1, loc_2 commands to dump the contents of the next location.

↑ This command may be used in conjunction with the loc and loc_1, loc_2 commands to dump the last location. The format of the command is

↑

***** This command may be used in conjunction with the loc and loc_1, loc_2 commands to dump the location whose address is contained in the location specified by loc. The format of the command is

*

MONITOR The MONITOR command turns the monitor display mode on and off (as does any explicit command). When the display mode is on, the current monitor is displayed. When the display mode is off, the dump is displayed. The format of the command is

MO[NITOR] [D[ISPLAY]]

where DISPLAY turns the monitor display mode on. Omission of DISPLAY turns the monitor display mode off.

loc = value This command places the specified value into the specified location (loc) of the running monitor. (The display mode must be on.) The format of the command is

loc = value

where

loc is the specified location.

value is the specified value.

MAP COMMANDS

These commands turn the map mode on and off. They work only with interactive commands and apply only to a particular user. The two map commands are

MAP

UNMAP

MAP The MAP command loads the map of the specified user if his JIT is in core. The format of the command is

MA[P], id

where id is the user identification assigned by the system. Dump output following a MAP command is assumed to be virtual addressed.

UNMAP The UNMAP command turns the mapping mode of operation off. The format of the command is

UN[MAP]

Dump output following an UNMAP command is assumed to be physical addressed.

SEARCH COMMANDS

Commands in this group allow core to be searched. The commands are

COMPARE

SMASK

SEARCH

COMPARE The COMPARE command compares dump locations between loc_1 and loc_2 with the running monitor, and outputs locations with nonequal contents. The format of the command is

CO[MPARE],loc_1,loc_2

SMASK The SMASK command sets the mask to the specified value. The format of the command is

SM[ASK],value

where value is a hexadecimal mask.

SEARCH The SEARCH command searches for and outputs all words between locations loc_1 and loc_2 that contain the specified value under the mask. The format of the command is

SE[ARCH],value,loc_1,loc_2

where

value is a hexadecimal value,

loc_1 is the beginning location and may be a hexadecimal number or an expression indicating a sum or difference of two hexadecimal numbers.

loc_2 is the ending location and may be a hexadecimal number or an expression indicating a sum or difference of two hexadecimal numbers.

OUTPUT COMMANDS

Commands in this group direct or format the output of Analyze. Four output commands are provided:

ROWS

LP

UC

PRINT

ROWS The ROWS command establishes the width of dump output. The format of the command is

ROWS value

where value is a number between 1 and 12. ROWS 1 would cause all hexadecimal dumps to be one word wide; ROWS 8 would cause the dumps to be eight words wide.

LP The LP command directs output from Analyze to the line printer. The format of the command is

LP[rows]

where rows indicates the dump width in number of words.

UC The UC command directs output from Analyze to the on-line terminal. The format of the command is

UC[rows]

where rows indicates the dump width in number of words.

PRINT The PRINT command closes the output symbiont file to allow output to the line printer without requiring a return to TEL. The format of the command is

PR[INT]

DEBUG COMMANDS

Commands in this group permit the use of Delta to facilitate monitor debugging. The three debug commands are

BF

DELTA

NODELTA

BF The BF command specifies the name of the boot file that represents the monitor being examined by Analyze. This enables the debugger Delta to read in the required symbol tables. If the BF command is not specified, the file M:MON in :SYS is the boot file that is assumed by default.

The form of the command is

BF fid

where fid is the file identification and is in the form

$$\text{name}\begin{bmatrix}[.[\text{account}].\text{password}]\\ .\text{account}\end{bmatrix}$$

DELTA The Delta command associates the debugger Delta with Analyze and gives control to Delta when the BREAK key is hit by the user. If the BF command has been issued, the Delta command ;S loads the global symbol table of the monitor root from the specified boot file. The Delta command name ;S loads the local symbol table of the module named. If the BF command was not executed, the file M:MON in :SYS is used to obtain the monitor symbol tables and the Delta commands apply to the running monitor being examined, not to the monitor in the boot file. The Delta command ;G is used to exit from Delta and to return control to Analyze.

The form of the DELTA command is

DEL[TA]

NODELTA The NODELTA command disassociates the debugger Delta from Analyze. The form of the command is

NO[DELTA]

MISCELLANEOUS COMMANDS

SYMBOLS This command creates an alphanumerically sorted monitor map by reading, sorting, and formatting the monitor's REF/DEF stack in the file MONSTK.:SYS. It is recommended that output be directed to the line printer for this command (see Figure 76).

The form of the command is

SY[MBOLS] [fid]

where fid is used to select symbols from a file and has the format

name [[. account].password | .account]

MONSTK.:SYS is the default.

DUMP This command causes a specified range of addresses to be dumped. The command's format is

DUMP loc_1, loc_2

Dump output following a MAP command is assumed to be virtual addressed; after an UNMAP command, physical addressed.

CLOSE This command causes the input dump file to be closed. The format is

CL[OSE]

A user should close a file prior to entering the monitor display mode.

HELP This command lists all Analyze commands and options, and gives a brief description of the purpose of each. The form of the HELP command is

HE[LP]

EXIT COMMANDS

There is one exit command: END

END The END command causes an exit from Analyze. The format of the command is

E[ND]

OUTPUT

The output produced by Analyze consists of a series of formatted monitor and user tables and the contents of registers existing at the time of the crash. The time and date information in the output page headings refer to the time at which the crash occurred.

Some of the output tables are chain type displays. That is, they are formed by starting at the head of a chained list and outputting that list until the tail of the chain is reached. If the tail and the last page in the chain do not agree, the following message is output:

TAIL ERROR

If the count differs from the number of pages in the chain, the following message is output:

COUNT ERROR

Examples of Analyze output with explanations of the information printed are given in the figures on the next several pages. In many cases the examples show only a portion of the display. The examples show the following types of displays:

Figure 39 - the contents of the registers at the time the dump was taken.

Figure 40 - the output for trap and interrupt locations. The trap and interrupt locations are those used by the associated XPSD instructions and are listed in Table 39.

Figure 41 - the core page in which the trap occurred, if a trap was the cause of the recovery.

Figure 42 - the user tables. This display includes the tables associated with each user that has a page count greater than zero. The meaning and source of items in this display are defined in Table 40.

Figure 43 - the user state chains which indicate the state of each user in the system.

Figure 44 - the contents of the swap tables. The meaning of each location in the table is defined in Table 41.

Figure 45 - the partition tables. Table 42 defines the heading in the partition tables display.

Figure 46 - the contents of the processor tables. Table 43 defines the meanings of the headings in the display.

Figure 47 - the monitor free page chain. The swapper page chain is formatted in the same manner as this figure. Usually there is no page chain data output.

Figure 48 - the user page chain display. This display indicates which pages and how many pages were being used by the various users resident in core.

Figure 49 - the processor page chain display. This display indicates which pages and how many pages were being used by the various processors resident in core.

Figure 50 - the ALLYCAT buffer adjustment tables. The headings used in this display are defined in Table 44.

Figure 51 - the actual physical memory allocation on a page-by-page basis. This display is a composite picture of the displays in Figures 47, 48, and 49, plus the resident monitor and its JIT, plus any unallocated pages.

Figure 52 - the contents of any unallocated pages.

Figure 53 - the I/O channel and device states. The display is separated into tables pertaining to each logical channel. For each channel, Analyze prints the channel information table (CIT), the device control tables (DCT) for devices on the channel, and the user I/O request queues on those devices. Table 45 defines the headings used in the display.

Figure 54 - the free queue entries which are used to contain user I/O requests for I/O devices defined in Figure 53.

Figure 55 - the channel information tables (CIT).

Figure 56 - the device control tables (DCT). Table 46 defines the meaning of the headings used in this display.

Figure 57 - the IOQ tables. Table 47 defines the meaning for the headings used in the IOQ tables display.

Figure 58 - the COC tables. This display includes the line table values for those lines having an associated user (determined by a non-zero value in LB:UN). Table 48 defines the headings used in the COC tables display.

Figure 59 - the resource allocation tables.

Figure 60 - the AVR tables. Table 49 defines the headings used in the AVR tables display.

Figure 61 - the contents of the incore error log buffers.

Figure 62 - the symbiont tables. The headings used in this display are defined in Table 50.

Figure 63 - the contents of the assigned CPOOLs and corresponding SPOOLs.

Figure 64 - the monitor JIT contents.

Figure 65 - the current user's JIT.

Figure 66 - the current user's TSTACK. TSTACK headings are defined in Table 51.

Figure 67 - the current user's AJIT (additional JIT).

Figure 68 - the current user's context area.

Figure 69 - the current user's physical pages.

Figure 70 - the monitor root.

Figure 71 - the RBBAT recovery file, which includes ghost communication buffers, the RBBAT environment, the RBBAT static data, and the RBBAT dynamic data. (Usually there is no dynamic data output.)

Figure 72 - the user identification. This display is a composite of all JITs in the MONDMP file.

Figure 73 - the patch file built by GHOST1 at system boot time.

Figure 74 - the current inswap and outswap users' core (if any). (This figure has the same format as Figure 75.)

Figure 75 - the current incore users' core.

Figure 76 - the symbol map.

Figure 77 - the Table of Contents for the Analyze dump.

```
REGISTERS:

   SCREECH CODE:  /E   SUB-CODE:  40

-- MONITOR HAS TRAPPED --

0        0000566C  00000043  00000002  00000000  00000057  00000000  00018C00  00000012
8        00000000  00007C93  000011D8  00003E5D  0000000C  44444400  00000800  00001324
```

Figure 39. Registers Display

```
TRAPS/INTERRUPTS:

INTRPT CONTENT  LAST PSD SAVED     CURRENT CONTENTS   NAME OF  NAME OF   NEW PSD
LOC    OF LOC   BY THIS TRAP       OF TRAPPED CELL    RECEIVER HANDLER   IS
------ -------- -----------------  -----------------  -------- --------  -----------------

  040 0FA0049A 000011D6 10000000 03007C04            NOPPSD   NOPPGM    00005C5E 17000000
  041 0FA004AE 00000000 00000000                     UNIMPPS  UNIMP     00005C9C 17000000
  042 0FA0049E 20C0CB3A 00000000 3520C00C   STW      STKLPSD  STKOVF    00005C68 17000000
  043 0FA004A2 00000000 00000000                     FIXOVPS  FIXOVF    00005C7E 17000000
  044 0FA004A6 00000000 00000000                     FLTFPSD  FLTFLT    00005C88 17000000
  045 0FA004AA A0E0C443 00000000 22200040   LI       DECFPSD  DECFLT    00005C92 17000000
  046 0FA004C4 00000000 00000000                     WDOGPSD  WDOGPGM   000038OD 17000000
  047 0FA004C4 00000000 00000000                     WDOGPSD  WDOGPGM   000038OD 17000000
  048 0FA004B2 20C0C1A2 00000000 0410C3F0   CAL1     CAL1PSD  CAL1P     00405C55 17000000
  049 0FA004B6 00C1C0D4 00000000 ED30000D   WD       CAL2PSD  CAL2XXX   00405C35 17000000
  04A 0FA004BA 00000000 00000000                     CAL3PSD  CAL3XXX   0040C5C3F 17000000
  04B 0FA004BE 00000000 00000000                     CAL4PSD  CAL4XXX   00405C49 17000000
  04C 0FA02104 00000000 00000000                     PARERPS  PARITYE   0000371A 17000000
  04D 0FA02110 00000000 00000000                     INSTXPS  INSTXCP   000037B3 17000000
  04E 0F8004E4 00000000 00000000                     LEE20              000004E9 07000000
  050 0FA004FA 00000001 00000001                     POWRON   BEGINON   00007FE8 07000000
  051 0FA004F2 00000000 00000000                     POWROFF  BEGINOF   000C7FDA 07000010
  056 0FA004E0 00000000 00000000                     PERPSD   MEMPAR    00007FBA 07000000
  057 0FA02114 00000000 00000000                     MEMFTPS  MEMFAUL   000037A5 17000000
  058 0FA004D0 00000000 00000000                     CLK1PSD  CLK1XXX   00005C33 17000000
  059 0FA004D4 00000000 00000000                     CLK2PSD  CLK2I     000055BA 17000000
  05A 0FA004D8 20002BD6 10000000 69802BD9   BCS      CLK3PSD  CLOCKI    00007E40 10000000
  05B 0FA004DC 80C0ED1D 00000000 D4F74040            CLK4PSD  CLK4      00003EF8 10000000
  05C 0F800492 3000597E 10000000 6AB05C89   BAL      IOPSD    IOINT     000C59C2 17000000
  05D 0F800496 30005926 10000000 6AB05C9F   BAL      OCPSD    OCINT     00005B0B 17000000
  060 0F80039E 20002BDA 10000000 64502BDA   BDR      CO:INO   COCIP     00002CA4 10000010
  061 0F800630 20002BDA 10000000 64502BDA   BDR                         00000634 17000010
  062 0F800628 60C0C85F 00000000 682AC867   BLE      CO:INN             0000062C 17000010
```

Figure 40. Trap and Interrupt Display

Table 39. Trap and Interrupt Locations for XPSD Instructions

Name	Meaning	Location of XPSD	Name	Meaning	Location of XPSD
NOPPSD	Nonallowed operation trap	X'40'	CAL4PSD	CAL4 instruction	X'4B'
UNIMP	Unimplemented instruction trap	X'41'	PONPSD	Power on	X'50'
STKLPSD	Stack overflow trap	X'42'	POFPSD	Power off	X'51'
FIXOV	Fixed-point arithmetic overflow	X'43'	PERPSD	Parity error	X'56'
FLTFPSD	Floating-point fault	X'44'	CLK1PSD	Counter 1 zero	X'58'
DECFPSD	Decimal arithmetic fault	X'45'	CLK2PSD	Counter 2 zero	X'59'
WDOGPSD	Watchdog timer runout	X'46'	CLK3PSD	Counter 3 zero	X'5A'
ABRTPSD	Unused trap location	X'47'	CLK4PSD	Counter 4 zero	X'5B'
CAL1PSD	CAL1 instruction	X'48'	IOPSD	Input/output interrupt	X'5C'
CAL2PSD	CAL2 instruction	X'49'	OCPSD	Control panel	X'5D'
CAL3PSD	CAL3 instruction	X'4A'	COCIN1	COC input interrupt	X'60'
			COCOUT1	COC output interrupt	X'61'

```
PAGE IN WHICH TRAP ØCCURRED

1000     68005925  09B08C4A  6A805004  08B08C4A  20300000  6820100C  22E00000  85E00003   •••••••  • •••• ••••••••••••5•••
1008     3530027E  20300100  32E00003  6800102A  6D000027  F230000A  32900008  209FFFFF   ••••••••••••••• •••2•••••••••
1010     6800062CC 22F1FE00  45E00000  68301018  32100000  82000000  69301012  68060C00   •••••1••••••••••••••2•••••••••
1018     62200000  35220000  68060001  09808C4A  6AB0101E  680062C7  60000037  2210027E   2•••••••••••••  ••••••G ••••••
1020     3200027E  68301028  6A301011  68001028  09808C4A  6AB0501A  08B08C4A  6800102A   ••••••• •••••••••  • •••• ••••
1028     46E0027E  85E0027E  6D000027  B240000A  3210000A  F230000A  E8000008  00000000   •••5••• •••2 ••••••2•••Y•••••••
1030     00001164  00000000  22201032  022000F0  08208C4A  35B6254C  2210000F  92020002   •••••••••••••••0••• •••<•••••••
1038     6AB05654  74000001  022000FC  0A208C4A  70200001  69F40010  22100081  7516254A    ••••••••••U••• ••••••••••••••
1040     08B08C4A  E8000008  3210000A  68007CBD  0F000001  D7C1C7C5  7BF3F140  40404040   ••• Y•••••••••••••••PAGE#31
1048     40404040  40404040  40404040  40404040  40404040  40404040  40404040  40404040
1050     07E2E8D4  C2C5D9D9  40404040  40404040  00001120  960001CC  02045000  16000634   •SYMBERR          •••••••••• •••••
1058     4040F040  4040D3D7  40404040  40404040  404040F1  40404040  40404040  40404040    0    LP            1
1060     40404040  4040D8D9  C5D4C1D2  C5404040  D47AE7E7  E740C1C2  D6D9E3E2  40404040         ØREMAKE   M:XXX ABØRTS
1068     40404040  40404040  40404040  40404040  40404040  40404040  40F1F5F3  40404040                           153
1070     40404040  40404040  40404040  40404040  40404040  40404040  00001096  96000008                       ••••••••
1078     02038800  160007F8  40404040  40404040  40404040  40404040  40404040  40404040   ••••••••8
1090*    40404040  40404040  40404040  40404040  40404040  40404040  40404040  40404040
1098     00001054  96000008  02033000  160007F8  4040F040  4040D3D7  40404040  40404040   •••••••••••••••8  0   LP
10A0     404040F1  C9C5E24B  C54B5C5C  5C5C5C5C  5C5C5C40  40404040  40404040  40404040     1IES•E•••••••••
10B0*    40404040  40404040  40404040  40404040  40404040  40404040  40404040  40404040
10B8     40404040  40404040  000010DC  96000008  02033800  160007F8  4040F040  4040D3D7          •••••••••••••••8  0   LP
10C0     40404040  40404040  404040F1  40404040  40404040  40404040  40404040  40404040        -     1
10D0*    40404040  40404040  40404040  40404040  40404040  40404040  40404040  40404040
10D8     40404040  40404040  40404040  40404040  00001076  96000008  0204A000  160007F8                 •••••••••••••••8
10E0     4040F040  4040D3D7  40404040  40404040  404040F1  40404040  40404040  40404040    0    LP            1
10F0*    40404040  40404040  40404040  40404040  40404040  40404040  40404040  40404040
10F8     40404040  40404040  40404040  40404040  40404040  40404040  000010BA  96000008                       ••••••••
1100     02071800  160007F8  4040F040  4040D3D7  40404040  40404040  404040F1  40404040   ••••••••8  0    LP           1
1118*    40404040  40404040  40404040  40404040  40404040  40404040  40404040  40404040
1120     00001142  96000574  12030800  1600028C  4040F040  4040D3D7  40404040  40404040   •••••••••••••••    0    LF
1128     404040F1  40404040  40404040  40404040  40404040  40404040  40404040  40404040    1
1138*    40404040  40404040  40404040  40404040  40404040  40404040  40404040  40404040
1140     40404040  40404040  00000000  96000574  12020800  1600028C  C5D740F2  F7684070         •••••••••••••••EP 27, '
```

Figure 41. Trap Page Display

USER TABLES:

USER	ST	BL	FL	FLG	FLG2	JIT	SWPI	HJIT	AJIT	PCT	ACP	APR	APØ	ASP	DB	ØV	TS	MF
43	SØN	0	57	90	0	53	0	0	1296	1	E	0	0	0	0	0	0	0
57	SØN	43	0	90	0	F3	0	0	1324	1	E	0	0	0	0	0	0	0
51	SCØM	56	0	301	0	55	0	C80	C50	2A	B	18	47	0	0	0	0	0
56	SCØM	0	51	211	0	BE	0	110C	0	F	A	16	0	0	0	0	0	0
3C	SCU	0	0	301	8	CA	0	E1A	E18	3E	B	18	41	0	0	0	0	0
2	SW	C	0	0	8	5E	0	4	0	E	0	D	0	0	0	0	0	0
3	SW	37	C	0	8	F1	0	C00	0	17	0	0	0	0	0	0	0	0
4	SW	0	6	1000	0	A9	1	16	14	14	C	1A	0	0	0	8	0	0
6	SW	4	37	0	18	CD	0	D1E	D1C	F	0	0	0	0	0	8	0	0
C	SW	3	2	0	8	BF	0	139A	0	C	A	0	0	0	0	8	0	0
37	SW	6	3	0	8	D8	0	CAE	0	9	A	0	0	0	0	8	0	0
1E	SIØC	0	0	201	0	43	0	CC0	CBE	12	A	19	3F	0	0	0	0	0
19	STØBØ	0	3F	0	8	CA	0	F0E	F0C	23	A	16	0	0	0	0	0	0
2E	STØBØ	3F	0	0	8	4B	0	1632	1630	E	A	15	0	0	0	0	0	0
3F	STØBØ	19	2E	0	8	79	0	158C	158A	2D	A	0	0	0	0	0	0	0
22	STIØ	30	44	0	8	9D	0	D26	D24	2D	A	1E	36	0	0	0	0	0
30	STIØ	0	22	0	8	F6	0	1594	1592	10	A	15	0	0	0	0	0	0
44	STIØ	22	0	0	8	6A	0	108E	0	11	A	0	0	0	0	0	0	0

Figure 42. User Tables Display

Table 40. User Table Headings

Heading	Source	Meaning
USER	-	Internal user number.
ST	UB:US	User's state.
BL	UB:BL	Link to previous user in same state.
FL	UB:FL	Link to next user in same state.
FLG	UH:FLG	User's flags.
FLG2	UH:FLG2	Exit control bits, miscellaneous control flags.
JIT	UB:JIT	Physical page address of user's JIT.
SWPI	-	Swap table index.
HJIT	UH:JIT	Track/sector address on the swapping RAD of user's JIT.
AJIT	UH:AJIT	Track/sector address of user's additional JIT.
PCT	UB:PCT	User's page count.
ACP	UB:ACP	Number of associated command processor.
APR	UB:APR	Number of associated processor's root.
APO	UB:APO	Number of associated processor's overlay.
ASP	UB:ASP	Number of associated special processor.
DB	UB:DB	Number of associated debugger.
OV	UB:OV	Number of associated overlay.
TS	UB:TS	Time remaining to user from last quantum.
MF	UB:MF	Number of I/O events outstanding.

```
USER STATE CHAINS:

STATE  STOB    HEAD 7 TAIL

STATE  SQA     HEAD 4 > 2 > 3 TAIL

STATE  SQFI    HEAD 5 TAIL

STATE  E:CEC   HEAD 8 > 9 > A > B > C > D > E > F > 10 >
                    11 > 12 > 13 > 14 > 15 > 16 > 17 > 18 >
                    19 > 1A > 1B > 1C > 1D > 1E > 1F > 20 >
                    21 > 22 > 23 > 24 > 25 > 26 > 27 > 28 >
                    29 > 6 TAIL
```

Figure 43. User State Chains Display

```
SWAP TABLES:

S:SIR=2CB0        S:HIR=1          S:SIP=0          #SWAP$DEV=0
S:CUN=3C          S:ISUN=0         S:CUIS=12        S:IDLF=229
SB:BSN=   C
SB:NP=   0
SB:FPN=   C

    M:SWAPD   MB:SDI   MB:SFC   MB:#RTRY   M:CLBGN   MH:CLEND
    2F0       7        3        5          122       1256
    1F0       6        5        5          10200     FFFF814B
```

Figure 44. Swap Tables Display

Table 41. Swap Table Terms

Location	Meaning	Location	Meaning
S:SIR	Swap in requests posted.	SB:NP	Number of in-swap processors.
S:HIR	High priority requests posted.	SB:PNL	In-swap processor numbers.
S:SIP	Swap-in progress flag.	SB:FPN	Number of freed processors.
#SWAP$DEV	Interrupt bypass count.	SB:FPL	List of freed processors.
S:CUN	Current user number.	M:SWAPD	Address of swap device.
S:ISUN	In-swap user number.	MB:SDI	DCT index.
S:CUIS	Count of users in system.	MB:SFC	Swap function code.
S:IDLF	Idle flag.	MB:#RTRY	Retry count.
SB:OSN	Number of out-swap users.	M:CLBGN	Beginning of current command list.
SB:OSUL	Out-swap user list.	MH:CLEND	End of current command list.
S:BECL	Beginning and end command list for each outswap user.		

```
PARTITION TABLES

# ACCOUNT  USR  FLG  WN    TOL    CUR   TL    TU    SID
1          C    FFFF8000 1388 11 1   138   138   0
2 :BCHSGEN 3C   2000 FA0   D      3     0     12C   282
3          C    0    FA0   11     6     0     1E    0
4          C    0    FA0   A      4     0     1E    0
5 :A00SGEN 51   2000 FA0   1E     12    0     3E7   263
6          0    0    FA0   48     10    0     3E7   0
7          0    0    FA0   27     2     0     3E7   0
8          0    0    FA0   53     7     0     A     0
9          0    0    FA0   44     8     0     A     0
A          0    0    FA0   41     B     0     A     0
B          0    0    FA0   13     5     0     A     0
C          0    0    FA0   4A     6     0     A     0
D          0    0    FA0   0      0     0     A     0
E          0    0    FA0   10     1     0     A     0
F          0    0    FA0   1      0     0     A     0
10         0    0    FA0   D      0     0     A     0
```

Figure 45. Partition Tables Display

Table 42. Partition Tables Headings

Heading	Source	Meaning
#	Calculated	Index to partition tables.
ACCOUNT	PLD:ACT	Current running account.
USR	PLB:USR	Number of users in partition.
FLG	PLH:FLG	Partition control flags.
QN	PLH:QN	Quantum time of partition.
TOL	PLH:TOL	Total jobs run in this partition.
CUR	PLH:CUR	Current jobs selected in this partition.
TL	PLH:TL	Lower time limit.
TU	PLH:TU	Upper time limit.
SID	PLH:SID	System ID.

PRØCESSØR TABLES:

P#	P:NAME	HPP	TPP	PSZ	DSZ	DCBSZ	PDA	DDA	UC	LNK	PVA	HVA	PC#	DC#	SA	TCB
1	KEYIN	0	FC	5	0	0	114	114	0	0	40	45	0	0	00008056	00000000
2	LDLNK	0	C8	2	0	0	110	110	0	0	40	42	0	0	00008035	00000000
3	DEBUG	0	4D	4	0	0	108	108	0	0	40	44	0	0	00008000	00000000
4	LTAPE	0	83	5	0	0	D0	D0	0	0	40	45	0	0	00008000	00000000
5	ØPEN	0	93	6	0	0	C4	C4	0	0	40	46	0	0	00008000	00000000
6	CLØSE	0	51	6	0	0	B8	B8	0	0	40	46	0	0	00008000	00000000
7	MUL	0	56	5	0	0	AE	AE	0	0	40	45	0	0	00008000	00000000
8	MISØV	0	58	5	0	0	A4	A4	0	0	40	45	0	0	00008000	00000000
9	IØDTYPR	0	45	3	0	0	9E	9E	0	0	40	43	0	0	00008000	00000000
A	TEL	0	F2	9	0	0	238	0	0	0	E1	EA	0	0	CC01C20C	00000000
B	CCI	0	6B	F	2	2	28C	250	0	0	62	71	0	0	8A00C5FC	00000000
C	GHØST1	0	D3	A	1A	2	8A	2B0	0	0	7A	84	0	0	8900F40C	00000000
D	ALLØCAT	0	9E	1	D	0	22	8	0	0	6D	6E	0	0	0000DA0C	00000000
E	LØGØN	0	8C	5	3	1	2B8	2B0	0	0	63	68	0	0	8F00C693	00000000
F	LINK	0	A4	7	1	1	2C6	2C2	0	0	E1	E8	0	0	4801CFD2	00000000
10	DELTA	0	D8	8	1	0	304	302	0	0	E1	E9	0	0	6001C868	00000000
11	:P00	0	0	14	0	0	314	0	0	0	E1	F5	0	0	50000000	00000000
12	:P11	0	8A	A	0	0	33C	0	0	0	E1	E8	0	0	50000000	00000000
13	LDEV	0	0	2	1	1	382	350	0	0	61	63	0	0	8000C285	00000000
14	EDIT	0	87	7	3	1	38E	386	0	0	63	6A	0	0	0000C60C	00000000
15	PCL	0	4F	10	2	2	3A4	39C	0	0	62	72	0	0	0000C40C	00000000
16	BASIC	AD	76	14	4	1	3CE	3C4	1	0	64	78	0	0	0000C068	00000000
17	METASYM	0	83	3	4	2	430	424	0	48	64	75	0	0	0000C822	00000000
18	LØADER	95	35	1	1	1	49E	49A	2	40	61	68	0	0	0800C100	00000000
19	FØRT	48	FF	7	4	2	51A	50E	1	3D	64	78	0	0	0000CE10	00000000
1A	FILL	0	7A	7	6	2	5AC	59C	0	0	66	6D	0	0	8800CC0C	00000000
1B	RUNNER	0	87	2	1	1	5BE	5BA	0	0	E1	E3	0	0	C001C20C	00000000
1C	BATCH	0	78	2	6	1	5D0	5C2	0	0	66	68	0	0	0000CC66	00000000
1D	TEXT	0	4F	14	3	1	60A	602	0	0	63	77	0	0	0000C60F	00000000
1E	APL	0	7A	13	3	2	63C	632	0	3B	63	7F	0	0	0000C206	00000000
1F	DRSP	0	0	6	2	1	6BA	6B4	0	0	62	68	0	0	0800C4C9	00000000

Figure 46. Processor Tables Display

Table 43. Processor Table Headings

Heading	Source	Meaning
HPP	PB:HPP	Head of processor's physical page chain.
P:NAME	P:NAME	Processor name.
TPP	PB:TPP	Tail of processor's physical page chain.
P#	-	Processor index number.
PSZ	PB:PSZ	Processor's procedure size in pages.
PC#	PB:PC#	Procedure cylinder number.
DSZ	PB:DSZ	Processor's initial data size in pages.
DC#	PB:DC#	Data cylinder number.
DCBSZ	PB:DCBSZ	Size in pages of DCB area.
PDA	PH:PDA	Disk address of procedure.
DDA	PH:DDA	Disk address of data and DCBs.
UC	PB:UC	Use count on processor.
LINK	PB:LNK	First overlay number for this processor.
PVA	PB:PVA	Virtual page address of the processor's procedure.
HVA	PB:HVA	First page available to the processor.
SA	P:TCB	Starting address and flags.
TCB	P:TCB	TCB address.

```
MONITOR (FREE) PAGE CHAIN:
                HEAD 69 > CC > D8 > C1 TAIL
                 TAIL C1
                COUNT 4
```

Figure 47. Monitor Free Page Chain Display

```
USER PAGE CHAINS:

USER#1E        HEAD AE > E5 > 73 > B8 > EE > BA > FD > 7D >
                    B9 > 90 > 7F > F4 > D2 > 96 > 98 > E0 >
                    42 > 43 TAIL
               TAIL 43
              COUNT 12

USER#3C        HEAD C9 > DF > A2 > A5 > C0 > 49 > 89 > 4B >
                    CF > 91 > 86 > C5 > A9 > 59 > EB > AC >
                    B1 > 4E > 65 > F8 > A8 > 66 > 5D > 60 >
                    C6 > AA > 86 > EC > F2 > 9B > B3 > 45 >
                    C3 > D9 > 6B > 72 > 58 > 64 > 8D > DB >
                    6D > 56 > 52 > EF > DC > 62 > 5B > 51 >
                    80 > C4 > FE > ED > FC > 50 > 8C > 92 >
                    BB > 61 > 53 > 93 > 81 > CA TAIL
               TAIL CA
              COUNT 3E

USER#51        HEAD E7 > 41 > F7 > F6 > 9C > F9 > B5 > 99 >
                    8B > 9F > 6C > D0 > FB > 75 > 8F > A6 >
                    9A > 4C > A3 > 4D > D7 > 78 > F3 > C8 >
                    47 > 6A > B4 > 9D > 77 > 82 > C7 > BD >
                    97 > A1 > 84 > DE > E1 > 87 > 54 > F1 >
                    F0 > 55 TAIL
               TAIL 55
              COUNT 2A

USER#56        HEAD CE > AB > D6 > 68 > 7E > 74 > 63 > 79 >
                    5C > FA > 7B > 85 > EA > E2 > BE TAIL
               TAIL BE
              COUNT F
```

Figure 48. User Page Chains Display

```
PROCESSOR PAGE CHAINS:

BASIC#16       HEAD AD > 4F > B0 > 83 > 8A > E8 > 6F > A4 >
                    B2 > A0 > 4A > 57 > 5E > 9E > BF > F5 >
                    A7 > BC > C2 > 76 TAIL
               TAIL 76
              COUNT 14

LOADER#18      HEAD 95 TAIL
               TAIL 95
              COUNT 1

FORT#19        HEAD 48 > 44 > CB > 7C > D4 > E4 > FF TAIL
               TAIL FF
              COUNT 7

PASS3F#3F      HEAD E9 > CD > 5A > 87 > DA > AF > D1 TAIL
               TAIL D1
              COUNT 7

PS1#41         HEAD 6E > 5F > 71 > 88 > 70 > E6 TAIL
               TAIL E6
              COUNT 6

FIN#47         HEAD D3 > 7A > E3 > D5 > 67 > 94 TAIL
               TAIL 94
              COUNT 6
```

Figure 49. Processor Page Chain Display

ALLYCAT TABLES:	TOP	BOTTOM	WORDCNT	TEMPBOT	BUFLAGS	ADJSTCNT	GRANULES AVAIL
RAD	49	21	28	21	0	0	12BC
PACK	7	42	45	42	0	0	8ECE
SYMB	42	9	39	39	0	0	73D
CYL	0	0	0	0	8000	0	0

Figure 50. ALLYCAT Buffer Adjustment Tables Display

Table 44. ALLYCAT Headings

Heading	Meaning
TOP	Top index into buffer.
BOTTOM	Bottom index into buffer.
WORDCNT	Number of disk addresses in buffer.
TEMPBOT	Set if ALLYCAT changing buffer.
BUFLAGS	Bit 0 = HGP empty, Bit 1 = buffer just filled, Bit 2 = buffer just emptied.
ADJSTCNT	Number of entries manipulated by ALLYCAT; may be either positive or negative.
GRANULES AVAIL	Total number of granules/cylinders remaining in system (in hexadecimal notation).

PHYSICAL MEMORY ALLOCATION:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V
1	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V
2	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V
3	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V	CP-V
4	CP-V	U#51	U#1E	U#1E	P#19	U#3C	JIT	U#51	P#19	U#3C	P#16	U#3C	U#51	U#51	U#3C	P#16
5	U#3C	U#3C	U#3C	U#3C	U#51	U#51	U#3C	P#16	U#3C	U#3C	P#3F	U#3C	U#56	U#3C	P#16	P#41
6	U#3C	U#3C	U#3C	U#56	U#3C	U#3C	U#3C	P#47	U#56	FREE	U#51	U#3C	U#51	U#3C	P#41	P#16
7	P#41	P#41	U#3C	U#1E	U#56	U#51	P#16	U#51	U#51	U#56	P#47	U#56	P#19	U#1E	U#56	U#1E
8	U#3C	U#3C	U#51	P#16	U#51	U#56	U#3C	P#3F	P#41	U#3C	P#16	U#51	U#3C	U#3C	*NO*	U#51
9	U#1E	U#3C	U#3C	U#3C	P#47	P#18	U#1E	U#51	U#1E	U#51	U#51	U#3C	U#51	U#51	P#16	U#51
A	P#16	U#51	U#3C	U#51	P#16	U#3C	U#51	P#16	U#3C	U#3C	U#3C	U#56	U#3C	P#16	U#1E	P#3F
B	P#16	U#3C	P#16	U#3C	U#51	U#51	U#3C	U#51	U#1E	U#1E	U#1E	U#3C	P#16	U#51	U#56	P#16
C	U#3C	FREE	P#16	U#3C	U#3C	U#3C	U#3C	U#51	U#51	U#3C	U#3C	P#19	FREE	P#3F	U#56	U#3C
D	U#51	P#3F	U#1E	P#47	P#19	P#47	U#56	U#51	FREE	U#3C	P#3F	U#3C	U#3C	*NO*	U#51	U#3C
E	U#1E	U#51	U#56	P#47	P#19	U#1E	P#41	U#51	P#16	P#3F	U#56	U#3C	U#3C	U#3C	U#1E	U#3C
F	U#51	U#51	U#3C	U#51	U#1E	P#16	U#51	U#51	U#3C	U#51	U#56	U#51	U#3C	U#1E	U#3C	P#19

Figure 51. Physical Memory Allocation Display

```
UNALLOCATED PAGES:

PAGE#DD

1BA00  0000C000 00854002 F140F0C3 C4C3F840 404040F4 F0F4F0F4 F0F4F040 40F4F0F4  •••••• •1 0LDC8    40404040  404
1BA08  F0C6F2C5 F24040C6 F2C6F4C3 F3C5F540 40C6F4C6 F1F4F0F4 F0404040 40F4F0F4  0F2E2  F2F4C3E5  F4F14040  404
1BA10  F0F4F0F4 F04040C6 F8C6F2C6 F6C6F240 40F0F0F2 C3C3F1C4 F74040C6 F0C6F5C6  04040  F8F2F6F2  002CC1C7  F0F5F
1BA18  F3C6F140 4040405C 40404040 4040F2E2 F2F4C3E5 F4F14040 40404040 F8F2F6F2  3F1    •    2S24CV41    8262
1BA20  4B4BC1D7 F0F5F3F1 5C404040 40400085 06044B4B 4B40F0C3 C4C4F040 404040C6  ••AP0531•    ••••••• 0CDD0  F
1BA28  F7C6F3C6 F1C6F840 40C6F7C6 F3C6F0C6 F74040C6 F0C6F1C6 F7C6F340 40F4F0F4  7F3F1F8  F7F3F0F7  F0F1F7F3  404
1BA30  F0F4F0F4 F0404040 40F4F0F4 F0F4F0F4 F04040F4 F0F4F0F4 F0F4F040 40C6F2C5  04040    40404040  40404040  F2E
1BA38  F2C6F2C6 F44040C3 F3C5F5C6 F4C6F140 4040405C F7F3F1F8 F7F3F0F7 F0F1F7F3  2F2F4  C3E5F4F1  •731873070173
1BA40  40404040 40404040 40404040 F2E2F2F4 C3E5F4F1 5C404040 40400085 06044B4B            2S24CV41•    ••••••
1BA48  4B40F0C3 C4C4F840 404040F4 F0F4F0F4 F0F4F040 40F4F0F4 F0C6F8C6 F24040C6  • 0CDD8   40404040  4040F8F2  F
1BA50  F6C6F4F0 F0F3C340 40C3F1C4 F7C6F0C6 F54C4040 40C6F3C6 F1C6F7C6 F34040C6  6F4003C  C1D7F0F5    F3F1F7F3  F
1BA58  F1C6F8C6 F7C6F340 40C6F0C6 F7C6F0C6 F14C40C6 F7C6F3F4 F0F4F040 4040405C  1F8F7F3  F0F7F0F1  F7F34040   •
1BA60  40404040 4040F8F2 F6F44B4B C1D7F0F5 F3F1F7F3 F1F8F7F3 F0F7F0F1 F7F34040        8264••AP0531731873070173
1BA68  5C404040 40400085 06044B4B 4B40F0C3 C4C5F040 404040F4 F0F4F0F4 F0F4F040  •      ••••••• 0CDE0    40404040
1BA70  40F4F0F4 F0F4F0F4 F04040F4 F0F4F0C6 F2C5F240 40C6F2C6 F4C3F3C5 F5404040   40404040  4040F2E2  F2F4C3E5
1BA78  40C6F4C6 F1F4F0F4 F04040F4 F0F4F0F4 F0F4F040 40C6F8C6 F2C6F6C6 F44040F0   F4F14040  40404040  F8F2F6F4  0
1BA80  FCF1C3C3 F1C4F740 4040405C 40404040 40404040 4040F2E2 F2F4C3E5 F4F14040  01CC1D7     •         2S24CV41
1BA88  40404040 F8F2F6F4 4B4BC1D7 5C404040 40400085 06044B4B 4B40F0C3 C4C5F840     8264••AP•     ••••••• 0CDE8
1BA90  404040C6 F0C6F5C6 F3C6F140 40C6F7C6 F3C6F1C6 F84040C6 F7C6F3C6 F0C6F740      F0F5F3F1  F7F3F1F8  F7F3F0F7
1BA98  40C6F0C6 F1C6F7C6 F3404040 40F4F0F4 F0F4F0F4 F04040F4 F0F4F0F4 F0F4F040   F0F1F7F3    40404040  40404040
1BAA0  40F4F0F4 F0F4F0F4 F04040C6 F2C5F2C6 F2C6F440 4040405C F0F5F3F1 F7F3F1F8   40404040  F2E2F2F4    •05317318
1BAA8  F7F3F0F7 F0F1F7F3 40404040 40404040 40404040 F2E2F2F4 5C404040 40400085  73070173             2S24•     ••
1BAB0  06044B4B 4B40F0C3 C4C6F040 404040C3 F3C5F5C6 F4C6F140 40F4F0F4 F0F4F0F4  ••••• 0CDF0    C3E5F4F1  40404
1BAB8  F04040F4 F0C3F2C3 F3C6F240 40C6F0C6 F0FCF0F1 C3404040 40C3F1C4 F7C6F0C6  0  40C2C3F2  F0F0001C    C1D7F0F
1BAC0  F04040C6 F0C6F0C6 F0C6F040 40C6F0C6 F8C6F7C6 F34040C6 F0C6F7C6 F0C6F140  0  F0F0F0F0  F0F8F7F3  F0F7F0F1
1BAC8  4040405C C3E5F4F1 40404040 40C2C3F2 F0F04B4B C1D7F0F0 F0F0F0F0 F0F8F7F3     •CV41      BC200••AP000000873
1BAD0  F0F7F0F1 5C404040 40400085 06044B4B 5C40F0C3 C4C6F840 404040C6 F7C6F3F4  0701•    ••••••• 0CDF8    F7F34
1BAD8  F0F4F040 40F4F0F4 F0F4F0F4 F04040F4 F0F4F0F4 F0F4F040 40F4F0F4 F0C6F2C5  040  40404040  40404040  4040F2E
1BAE0  F2404040 40C6F2C6 F4C3F3C5 F54040C6 F4C6F1F4 F0F4F040 40F4F0F4 F0F4F0C3  2    F2F4C3E5  F4F14040  40404C
1BAE8  F24040C3 F3C6F2C6 F0C6F640 4040405C F7F34040 40404040 40404040 4040F2E2  2  C3F2F0F6     •73             2S
```

Figure 52. Unallocated Pages Display

```
I/O CHANNEL/DEVICE STATES:

  I/O  CHANNEL 1
CIT3 CIT4 CIT5
0    0    0
DEVICE ADDR CX  8IUTS    BPwXKCSB QX AI8      TDV
TYA01   1    1  11000000 00010000 5  00100001 0000108D00000000

  I/O  CHANNEL 2
CIT3 CIT4 CIT5
0    0    0
DEVICE ADDR CX  8IUTS    BPwXKCSB QX AI8      TDV
LPA02   2    2  10000000 00010000 5  00900002 0000108600800000

  I/O  CHANNEL 3
CIT3 CIT4 CIT5
0    0    0
DEVICE ADDR CX  8IUTS    BPwXKCSB QX AI8      TDV
LPA0F   F    3  10000000 00010000 3  0090000F 0000108900800000

  I/O  CHANNEL 4
CIT3 CIT4 CIT5
0    0    0
DEVICE ADDR CX  8IUTS    BPwXKCSB QX AI8      TDV
CRA03   3    4  01000000 00010000 F  00900003 000010900080003C

  I/O  CHANNEL 5
CIT3 CIT4 CIT5
0    0    0
DEVICE ADDR CX  8IUTS    BPwXKCSB QX AI8      TDV
CPA04   4    5  10000000 00010000 1  01000004 0000109400000000

  I/O  CHANNEL 6
CIT3 CIT4 CIT5
```

Figure 53. I/O Channel and Device States Display

Table 45. I/O Table Headings

Heading	Meaning	Heading	Meaning
CIT3-5	Channel Information Tables 3-5	BPWXKCSB	From DCT5 - Set bits indicate: B Device busy P Clean-up pending W Wait until done X Data transfer K Wait for key-in C Control task S SIO while manual B BIN mode
DEVICE	yyndd for this device		
ADDR	Hardware address		
CX	Channel index		
OIDTS	From DCT3 - Set bits indicate: O output I input D down T timed out S SIO reject	QX	I/O queue index
		AIO	Last AIO status
		TDV	Last TDV status

FREE QUEUE ENTRYS:

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
0	1	0	0	0	0	12	6EC00	400	A	FF	1FF7F02	7C90	FA	0
0	5	80	C	0	22013086	A	48000	800	3	3	D060000	0	FE	1E
0	E	80	C	0	0	7	47400	400	A	A	B030000	7C90	FA	0
0	12	80	0	0	0	7	47000	400	A	A	13210000	7C90	FA	0
0	10	80	1	7	0	11	60800	800	A	A	11E0002	69A6	FE	0
1	D	80	1	1	0	A	32800	800	A	A	D060000	69A6	FE	0
0	C	80	0	5	220188FE	11	8000085D	2	A	A	430D04	0	FE	3F
0	14	80	C	0	0	6	40000091	1	0	0	3525	353E	10	0
E	4	80	0	5	9EFC	17	7C000	800	A	A	9D0600	0	FE	56
5	B	80	0	5	2200C87C	11	8000082A	2	8	8	17E0700	0	FE	3C
0	15	80	C	5	2201EEE6	17	800008B2	2	8	8	510A00	0	FE	1E
0	7	80	C	5	2201EEE6	17	8000082A	2	8	8	500701	0	FE	1E
0	17	80	1	7	260110E6	17	2F800	800	8	8	681103	0	FE	3C
0	11	80	1	7	260110E6	17	3A000	800	8	8	681105	0	FE	3C
0	16	80	0	0	0	6	40008B00	1	0	0	3525	353E	10	0
0	6	80	C	0	0	6	4000D6DC	1	0	0	3525	353E	10	0
0	F	80	1	1	0	D	53000	800	30	30	0	D22C	FE	FF
0	A	80	C	0	0	7	20400	400	A	A	14E60000	7C90	FA	0
7	2	80	1	1	0	D	5C800	800	30	30	0	1942C	FE	FF
12	13	80	1	1	0	D	78800	800	30	30	0	1942C	FE	FF
14	8	80	1	7	401192A	13	20800	800	A	A	190C04	0	FE	1B
0	9	8C	1	7	0	17	30800	800	A	A	360D09	69A6	FE	0
0	18	80	1	7	0	17	34800	800	A	A	360E06	69A6	FE	0
0	19	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 54. Free Queue Entries Display

20:50 SEP 27,'73   SITE ID:7E   VERSION:A00   FILE#6   PAGE#27

CHANNEL INFORMATION TABLE:

CHANNEL INFORMATION

#	1	2	3	4	5	6
00	00	00	00	00	00	00
00	00	00	00	00	00	00
00	00	00	00	00	00	00
00	00	00	00	00	00	00
00	00	00	00	00	00	00
00	00	00	00	00	00	00
00	00	00	00	00	00	00
00	00	00	00	00	00	00
00	00	00	00	00	00	00
00	00	00	00	00	00	00

Figure 55. Channel Information Table Display

DEVICE CONTROL TABLES:

#	1	1A	1P	2	3		5	6	7	8	9	10	11	12	13<--	-->13	14	15	17	18	19	20	21	22	23
#	DEV	PRI	ALT	CI	LG	TY	ST	Q#	CDWAD	INITE	CLENE	DAC	TIMEOUT	AIB/IBS	TDV	STATUS	DE	IS	HAND	IO	ACIT	LP	TYPE	DP	DISP
01	001	C01	001	01	C0	TY	10	05	0211A	0294E	02950	5F6	0000883	00100001	0000108D	00000000	00	00	00000	04	0030	20	1000	00	0000
02	003	003	003	04	40	CR	10	0F	02120	0299F	029A9	008	0000089	00900003	00001090	0080003C	00	00	00000	01	0030	20	0080	00	0000
03	004	004	004	05	80	CP	10	01	02122	02AE2	02B16	186	000046D	01000004	00001094	00000000	00	00	00100	01	0030	20	1000	00	0000
04	002	002	002	02	80	LP	10	05	0216C	029DC	029ED	5C4	0000886	00900002	000010B6	00800000	00	00	00300	01	0030	20	1080	00	0000
05	00F	C0F	00F	03	80	LP	10	03	02172	029DC	029ED	780	0000886	0090000F	000010B9	00800000	00	00	00300	01	0030	20	1080	00	0000
06	1F0	1F0	1F0	06	C0	DC	10	03	02178	02982	02984	628	0000886	001001F0	0000122C	00000000	00	00	00000	01	0030	20	1000	01	0000
07	2F0	2F0	2F0	07	C0	DC	10	05	0217E	02982	02984	0A2	0000886	001002F0	000010C0	00000000	00	00	00000	01	0030	20	1000	02	0008
08	2F1	2F1	2F1	07	C0	DC	10	01	02184	02982	02984	A4E	0000886	001002F1	000010C3	00000000	00	00	00000	01	0030	20	1000	02	0010
09	2F2	2F2	2F2	07	C0	DC	10	01	0218A	02982	02984	6E8	0000886	001002F2	000010C6	00000000	00	00	00000	01	0030	20	1000	02	0018
0A	2F3.	2F3	2F3	07	C0	DC	10	01	02190	02982	02984	7C0	0000886	001002F3	000010C9	00000000	00	00	00000	01	0030	20	1000	02	0020
0B	2F4	2F4	2F4	07	C0	DC	10	0E	02196	02982	02984	21E	0000886	001002F4	000010CC	00000000	00	00	00000	01	0030	20	1000	02	0028
0C	080	C80	080	08	C0	9T	00	12	0219C	02B30	02B37	2A0	000062D	40000080	000010CE	44000001	00	00	00A00	1E	0030	20	1000	00	0000
0D	081	C81	081	08	C0	9T	00	0C	021A4	02B30	02B37	150	00001DD	40000081	000010DA	44000001	00	00	00A00	1E	0030	20	1000	00	0000
0E	082	C82	082	08	C0	9T	00	00	021AC	02B30	02B37	000	0000000	00000000	00000000	00000000	00	00	00000	00	0000	00	0000	00	0000
0F	083	C83	083	08	C0	9T	00	12	021B4	02B30	02B37	5E6	00001EC	40000083	000010DA	44000001	00	00	00A00	1E	0030	20	1000	00	0000
10	084	C84	084	08	C0	9T	00	00	021BC	02B30	02B37	000	0000000	00000000	00000000	00000000	00	00	00000	00	0000	00	0000	00	0000
11	0E0	CE0	0E0	08	C0	DP	10	12	021C4	02B78	02B9F	CDC	0000886	041000E0	000010E2	04000000	00	00	00103	01	0030	20	1000	03	0030
12	0E1	CE1	0E1	08	C0	DP	00	03	021D0	02B78	02B9F	786	0000887	040800E1	000010E8	24000000	00	00	00905	01	0070	20	1800	03	0038
13	0E2	CE2	0E2	08	C0	DP	10	03	021DC	02B78	02B9F	730	0000886	041000E2	000010EE	04000000	00	00	00000	01	0030	20	1000	03	0040
14	0E3	CE3	0E3	08	C0	DP	10	12	021E8	02B78	02B9F	764	0000884	041000E3	000010F4	04000000	00	00	00000	01	0030	20	1000	03	0048
15	000	C00	000	08	C0	DP	00	00	021F4	02B78	02B9F	000	0000000	00000000	00000000	00000000	00	00	00000	00	0000	00	0000	04	0050
16	0E1	CE1	0E1	08	C0	DP	00	00	02200	02B78	02B9F	000	0000000	00000000	00000000	00000000	00	00	00000	00	0000	00	0000	04	0058
17	0D0	0D0	0D0	08	C0	DP	10	03	0220C	02C04	02C28	2A8	0000886	001000D0	00001106	00000000	00	00	00000	01	0030	20	1000	05	0060
18	0D1	CD1	0D1	08	C0	DP	10	03	02218	02C04	02C28	288	0000886	001000D1	0000110C	00000000	00	00	00000	01	0030	20	1000	05	0068
19	010	C10	010	09	C0	ME	00	00	02224	02C88	02C88	000	0000000	00000000	00000000	00000000	00	00	00000	00	0000	00	0000	00	0000
1A	011	C11	011	09	C0	ME	00	00	0222A	02C88	02C88	000	0000000	00000000	00000000	00000000	00	00	00000	00	0000	00	0000	00	0000

Figure 56. Device Control Tables Display

Table 46. Device Control Table Headings

Heading	Meaning
#	DCT number.
DEV	Active I/O address.
PRI	Primary I/O address.
ALT	Alternate I/O address.
CI	Channel (CIT) index.
LG	I/O legality: 11 = in and out 10 = out only 01 = in only
TY	Type mnemonic.
ST	State of device.
Q#	IOQ index.
CDWAD	Command doubleword address (WA resolution).
INITE	Handler preprocessor word address.
CLENE	Handler postprocessor word address.
DAC	Device activity counter.
TIMEOUT	Value to match against I/O clock.
AIO/IOS	AIO status word.
TDV STATUS	TDV status doubleword.
DE	Dedicated device flag.
IS	7446 table.
HAND	Handler function flags (first 8 bits contain retry function code; the second 8 bits contain the follow on code).
TO	7446 flags.
ACIT	Alternate CIT index.
LP	7446 flags.
TYPE	7446 flags.
DP	Disk flag.
DISP	Heading Granule Pool (HGP) displacement if disk.

```
IOQ TABLES:

       1   2           7    3   4  5  6     6          8         9  10  11         12 13<--           -->13   14   15
 #   BAK FWD  DCT# & MNE STAT FCN CMDS DCBAD     BUF/TIM/CDW NRA NRT    RAD AD E A ADR    E A INFO PRIO USER
--   --- ---  ---------- ---- -------- -----  -------------- --- --- -------- ----------------- ---- -----
01    00  05  CA   DCLF3   80 00 00 22 13086 00048000  0800  03  03 00060000 00000000 00000022  FE   1E
02    12  13  0D   9TA81   80 01 01 00 00000 00078800  0800  30  30 00000000 0001942C 00000006  FE   FF
03    00  01  12   DPLD1   00 00 00 00 00000 0006EC00  0400  0A  FF 01FF7F02 00007C90 00001108  FA   00
04    05  0B  11   DPLD0   80 00 05 22 0C87C 8000082A  0002  08  08 017E0700 00000000 00000022  FE   3C
05    00  0E  07   DCLF0   80 00 00 00 00000 00047400  0400  0A  0A 08030000 00007C90 00001200  FA   00
06    0D  0F  0D   9TA81   80 01 01 00 00000 00053000  0800  30  30 00000000 0000D22C 00000006  FE   FF
07    00  17  17   DPAD0   80 01 07 26 110E6 0002F800  0800  08  08 00681103 00000000 00000026  FE   3C
08    00  09  17   DPAD0   80 01 07 00 00000 00030800  0800  0A  0A 00360D09 000069A6 0000006A  FE   00
09    00  18  17   DPAD0   80 01 07 00 00000 00034800  0800  0A  0A 00360E06 000069A6 0000006A  FE   00
0A    07  02  0D   9TA81   80 01 01 00 00000 0005C800  0800  30  30 00000000 0001942C 00000005  FE   FF
0B    00  15  17   DPAD0   80 00 05 22 1EEE6 800008B2  0002  08  08 00510A00 00000000 00000022  FE   1E
0C    00  14  06   DCBF0   80 00 00 00 00000 40000091  0001  00  00 00003525 0000353E 004F3C14  10   00
0D    00  0C  11   DPLD0   80 00 05 22 1B8FE 6000085D  0002  0A  0A 00430D04 00000000 00000022  FE   3F
0E    00  12  07   DCLF0   80 00 00 00 00000 00047000  0400  0A  0A 13210000 00007C90 00001186  FA   00
0F    00  0A  07   DCLF0   80 00 00 00 00000 00020400  0400  0A  0A 14E60000 00007C90 00001186  FA   00
10    01  0D  CA   DCLF3   80 01 01 00 00000 00032800  0800  0A  0A 00060000 000069A6 000000C1  FE   0C
11    00  16  06   DCBF0   80 00 00 00 00000 40008800  0001  00  00 00003525 0000353E 00000006  10   0C
12    00  10  11   DPLD0   80 01 07 00 00000 00060800  0800  0A  0A 011E0002 000069A6 000000F4  FE   00
13    14  08  13   DPLD2   80 01 07 04 1192A 00020800  0800  0A  0A 00190C04 00000000 00000800  FE   1B
14    0E  04  17   DPAD0   80 00 05 00 09EFC 0007C000  0800  0A  0A 009D0600 00000000 0000687B  FE   56
15    00  07  17   DPAD0   80 00 05 22 1EEE6 8000082A  0002  08  08 00500701 00000000 00000022  FE   1E
16    00  06  06   DCBF0   80 00 00 00 00000 400006DC  0001  C0  00 00003525 0000353E 00000000  10   00
17    00  11  17   DPAD0   80 01 07 26 110E6 0003A000  0800  08  08 00681105 00000000 00000026  FE   3C
18    00  19  C0 .......   00 00 00 00 00000 00000000  0000  00  00 00000000 00000000 00000000  00   00
19    00  00  C0 .......   00 00 00 00 00000 00000000  0000  00  00 00000000 00000000 00000000  00   00
```

Figure 57. IOQ Tables Display

Table 47. IOQ Table Headings

Heading	Meaning
#	IOQ table number.
BAK	Back link to next entry.
FWD	Forward link to next entry.
DCT#	DCT index.
MNE	TEXT name of device from SYSGEN.
STAT	Software status.
FCN	Original function code (IOQ4).
CODS	Current function code (IOQ5).
DCBAD	DCB word address (if any).
BUF	Buffer BA if bit 0 and 1 reset; CDW DA if bit 1 set (swapper); CDW DA if bit 0 set (other).
TIM	Number of timeout increments.
CDW	Number of commands used if IOQ8 bit 0 or 1 set.
NRA	Original number of recovery tries.
NRT	Remaining number of recovery tries.
RAD AD	Disk address.
E A ADR	End action word address.
E A INFO	One word to return to end action receiver.
PRIO	Priority of this event.
USER	User number of I/O requester.

```
COC TABLES:
                                                       MODE BYTES   - - -INPUT TABLES- - -    OUTPUT TABLES
LINE  USER  TYPE   EOMTIME  BUFCNT  CPOS  RSZ          2  3  4     TL    II   IR  ARSZ CPI   OI   OR   OC
  0    19   TTY33   CD99      3       1    84      88 20  0  0   8000  0    0    2    1   1AF  113  2A
  F    56   TTY33   E169      0       1    84      88 20  0  0   8000  0    0    B    1   92   0    0
 10    C    TTY33   6BCF      0       1    8       88 20  0  0   8000  0    0    1    1   328  0    0
 11    37   TTY33   958D      0      11    50      68 20 80  0   8000  0    0    1    1   246  0    0
 12    3F   TTY33   3311      5       1    50      88 20 80  0   8000  0    C    0    1   2EF  168  41
 13    44   TTY33   B61       1       F    50      98 20  0  0   8000  298  292  A    5   295  0    0
 26    22   SAPLU   BA3       1       A    64      10 16  0  9   8000  1C4  1C2  3    7   118  0    0
 3D    30   TTY33   B6F       0       2    8C      98 20  0  2B  8000  0    0    C    2   11B  0    0
 3E    1F   TTY37   BE7B      0       1    50      88 20 80 13   8000  0    0    3    1   1CC  0    0
 40    2E   TTY33   C997      6       1    50      88 20 80  0   8000  0    0    13   1   18D  318  4D
```

Figure 58. COC Tables Display

Table 48. COC Line Table Headings

Heading	Source	Meaning
LINE	Calculated	Line number.
USER	LB:UN	Associated internal user.
TYPE	COCTERM	Terminal type.
EOMTIME	EOMTIME	End of message time for a read.
BUFCNT	BUFCNT	Number of buffers in use for line.
CPOS	CPOS	Current carriage position.
RSZ	RSZ	Record size requested by user while read is pending.
MODE BYTES	MODE-MODE4	Terminal mode indicators.
TL	TL	Pointer to tab buffer.
II	COCII	Input insertion pointer for line.
IR	COCIR	Input removal pointer for line.
ARSZ	ARSZ	Accumulated record size while read is pending.
CPI	CPI	Initial carriage position for a read.
OI	COCOI	Output insertion pointer for line.
OR	COCOR	Output removal pointer for line.
OC	COCOC	Count of characters pending output.

```
RESOURCE ALLOCATION TABLES:

                             CURRENT                 DEFAULT                AVAILABLE               MAXIMUM
NAME  TYPE  TOTAL     BATCH ONLINE GHOST    BATCH ONLINE GHOST    BATCH ONLINE GHOST    BATCH ONLINE GHOST
----  ----- -----     ----- ------ -----    ----- ------ -----    ----- ------ -----    ----- ------ -----

 C9    FF   7FFF      0060   0000  0000      20     14    FF      00A4   0014  7FFF      40     14    FF
 9T    08   0005      0001   0000  0000      00     00    02      0005   0004  0004      05     04    04
 7T    09   0000      0000   0000  0000      00     00    00      0000   0000  0000      00     00    00
 SP    08   0000      0000   0000  0000      00     00    00      0000   0000  0000      00     00    00
```

Figure 59. Resource Allocation Tables Display

```
AVR TABLES:

  MAGNETIC TAPES

  SER# PUB POS AVR SCR HLD PTL UPL OPN NOU TPOS    USER SOLICIT DCBS+USERS
  ---- --- --- --- --- --- --- --- --- --- ----    ---- ------- ----------

     0   0   0   0   0   0   0   0   0  00 0000     000   00       000C
     0   0   0   0   0   0   0   0   0  00 0000     000   00       0000
     0   0   0   0   0   0   0   0   0  00 0000     000   00       0000
     0   0   0   0   0   0   0   0   0  00 0000     000   00       0000
     0   0   0   0   0   0   0   0   0  00 0000     000   00       0000
  DISK PACKS

  SER# PUB     AVR INI VER MTD PRIM     NOU HGPDISP USER SOLICIT DCBS+USERS
  ---- ---     --- --- --- --- ----     --- ------- ---- ------- ----------

     0   1       0   0   0   0   0      00  0030     000   00       0000
     0   1       0   0   0   0   0      00  0038     000   00       0000
     0   1       0   0   0   0   0      00  0040     000   00       0000
     0   1       0   0   0   0   0      00  0048     000   00       0000
     0   1       0   0   0   0   0      00  0050     000   00       0000
     0   1       0   0   0   0   0      00  0058     000   00       0000
     0   1       0   0   0   0   0      00  0060     000   00       0000
     0   1       0   0   0   0   0      00  0068     000   00       0000
```

Figure 60. AVR Tables Display

Table 49. AVR Table Headings

Heading	Meaning
SER#	Serial number of tape or pack.
PUB	Set if public.
POS	Set if positioned.
AVR	Set if AVRed.
SCR	Set if scratch tape.
HLD	Set if held.
PTL	Set if positioned to label.
UPL	Set if user positioned label.
OPN	Set if open.
NOU	Number of users.
TPOS	Record count.
USER	User number.
SOLICIT	Index to special AVR tables.
INI	Set if volume initialized.
VER	Set if volume verified.
MTD	Set if mounted.
PRIM	Set if primary volume of private set of volumes.
HGPDISP	Displacement from HGP.

```
IN CORE ERROR LOG DATA:

(MOST RECENT ENTRY FIRST)

 CORE
 ADDR
                  TIME OF ENTRY: 20:50

5C6     150BC4D7  F2F0F5F0  040800E1  00207000  000010E8  24000000  83004A90  04000004  ..DP2050..............Y...... .....
5CE     18000A0A  FFFF8781  01FF7F02                                                    .......... .
                  TIME OF ENTRY: 20:50

5BB     150BC4D7  F2F0F5F0  040800E1  00207000  000010E8  24000000  83004A9C  04000004  ..DP2050............Y...... .....
5C3     18000A0A  FFFF8770  01FF7F02                                                    ........'.. .
                  TIME OF ENTRY: 20:50

5B0     150BC4D7  F2F0F5F0  040800E1  00207000  000010E8  24000000  83004A90  04000004  ..DP2050............Y...... .....
5B8     18000A0A  FFFF8779  01FF7F02                                                    .......... .
                  TIME OF ENTRY: 20:50

5A5     150BC4D7  F2F0F5F0  041900E1  00207000  000010E8  44000400  0206EC00  16000400  ..DP2050............Y...........
5AD     18000A0A  FFFF8773  01141002                                                    ............
                  TIME OF ENTRY: 20:14

597     1A0E75FF  F2F0F1F4  03D1C9E3  40404040  4040404C  02000202  7AC4F0F0  C2D64040  ....2014.JIT          ....:D00B0
59F     03000002  00000000  7AC4F0F0  C2D64040  0012364A  00010001                      ........:D00B0  ... ....
```

Figure 61. Core Error Log Buffers Display

```
OUTPUT SYMBIONT TABLES:

#   SQUE  SNDDX&TYPE    SSTAT   SSIG   SRET       SCNTXT  SYMX  TYP  LNK  FLAG  SUSP  QUE
--  ----  ----------    -----   ----   --------   ------  ----  ---  ---  ----  ----  ---
01  00    02  LMA03     00      .=00   00007AEA   00000   01
02  00    04  LPA02     01      I=00   00007CE0   01200   02
03  00    05  LPA0F     01      I=00   000240F2   011D8   02
04  00    03  CPA04     00      .=00   00000007   00000   02
05  00    00  .......   00      .=00   00000000   00000   03
06  02    00  .......   01      I=00   00007CE0   01188   02

          SQHD        SQTL
          --------    --------
          00000600    00000200
```

Figure 62. Symbiont Tables Display

Table 50. Symbiont Table Headings

Heading	Meaning	Heading	Meaning
#	Index number of table.	SCNTXT	Context block doubleword address displacement.
SQUE	Symbiont queue chain.	SYMX	Symbiont index: 1 = input 2 = output
SNDDX	DCT index of symbiont device.	TYP	Device type.
TYPE	TEXT name of symbiont device from SYSGEN.	LNK	Remote chain.
SSTAT	Symbiont Status: 0 = input symbiont 1 = output symbiont	FLAG	Remote flags.
		SUSP	Suspend bit for IRBT.
SSIG	Symbiont signal character (e.g., L, Q, etc.).	QUE	IOQ index for IRBT.
SRET	Symbiont return when activated from chain.	SQHD	Symbiont queue chain head.
		SQTL	Symbiont queue chain tail.

```
***ASSIGNED CPOOLS

1188     C6007C04 00800000 00070E52 00070E59 01070E5A 00FA0A00 00011C00 00000400  ..@................ ...........
1190     00070E59 00007C90 00FF0300 00000004 00000000 00000006 00007CF4 12000C00  ......@..................@4.....
1198     01000000 03030303 01010303 00000000 00000001 00000431 00000431 00000000  ................................
11A0     00000000 00000000 00000000 00000000 0100001D 00070E58 00000000 000056D9  ...............................R
11A8     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000  ................................

11D8     03007C04 00000006 00000000 001281C2 01070C8C 00FA0A00 00018B00 00000400  ..@............B................
11E0     001281C2 00007C90 03FF0305 00000004 00000079 00000003 00007CF4 07000257  ...B..@..................@4.....
11E8     08000000 03030303 01010303 00000004 001387C2 000000E0 0000071F 0000002D  .....................B..........
11F0     00000000 00000000 00000308 0018AB81 0100001D 00000000 00000000 000056D9  ...............................R
11F8     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000  ................................

1200     02007C04 00000006 0007070C 00070843 01145484 00FA0A00 00011D00 00000400  ..@.............................
1208     00070843 00007C90 03FF0304 00000004 00000079 00000002 00007CF4 07000261  ......@..................@4..../
1210     01000000 03030303 01010303 00000081 00070842 00000051 00000281 00000012  .........................1......
1218     00000000 00000000 00000314 00070842 0100001D 00000000 00000000 000056D9  ...............................R
1220     00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000  ................................

***AND THEIR SPOOLS

11C00    00070E52 00850641 F140C8F0 F1404040 F1F97AF1 F24040E2 C5D740F2 F768407D  ........1 H01   19112  SEP 27,
11C08    F7F34040 40404040 40404040 40404040 40404040 40404040 40404040 40404040  73
11C10    40404040 40404040 40404040 40404040 4040404C 40404040 40404040 40404040                234
11C18    40404040 40404040 40404040 40404040 40F2F3F4 40404040 40404040 40404040                ....5        4051
11C20    40404040 40404040 40404040 40007906 04F54040 40404040 4040F4F0 F5F14040
11C28    40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040   PAGE
11C30    404040D7 C1C7C540 40404040 40404040 4040404C 40404040 40404040 40404040
11C38    40404040 40404040 40404040 40404040 40404040 40404040 4040F4F0 F5F24040                ....6        4052
11C40    40404040 40404040 40404040 40007906 04F64040 40404040 4040C9D5 E2C5D9E3        06 00D8B           INSERT
11C48    40404040 404040F0 F640F0F0 C4F8C240 40404040 40404040 40404040 4040C9D5  LB EQU      *                IN
11C50    D3C240C5 D8E44040 40404040 58404040 40404040 40404040 40404040 40404040  SERT CORE LIBRARY NAME
11C58    E2C5D9E3 40C3D6D9 C540D3C9 C2D9C1D9 E840D5C1 D4C54040 40404040 F5F34C40                ....7        4053
11C60    40404040 40404040 40404040 40007906 04F74040 40404040 4040F4F0 40404040   06 0008B    49300E34
11C68    40F0F640 F0F0C4F8 C2404040 40F4F9F3 F0F0C5F3 F4404040 40404040 4040C9D5     BR,R3     =X'037A0C00'     IN
11C70    404040D6 D96BD9F3 40404040 7EE77DF0 F3F7C1F0 F0F0F07D 40404040 40404C40  SERT COUNT :
11C78    E2C5D9E3 40C3D6E4 D5E3407A 40404040 40404040 40404040 40404040 40404C40                ....8        4054
11C80    40404040 4040404C 40404040 40007906 04F84040 40404040 4040F4F0 F5F44040   06 00D8C    35300188 01
11C88    40F0F640 F0F0C4F8 C3404040 40F3F5F3 F0F0F1F8 F840F0F1 40404040 4040C6C9     STW,R3    OPNCORL1          FI
11C90    4C4040E2 E3E66BD9 F3404040 D6D7D5C3 D6D9D3F1 40404040 40404040 F0F37AD7  LE NAME OF CORE LIB RFDFSTK(03:P
11C98    D3C540D5 C1D4C540 D6C640C3 D6D9C540 D3C9C240 D9C6C4C6 E2E3D24D F5F54040  N)             ....9        4055
11CA0    D5D54040 40404040 40404040 40007906 04F94040 40404040 4040F4F0 F5F54040   06 00D8D    3530017F 01
11CA8    40F0F640 F0F0C4F8 C4404040 40F3F5F3 F0F0F1F7 C640F0F1 40404040 40404040     STW,R3    PNSIZE
11CB0    404040E2 E3E66BD9 F3404040 D7D5E2C9 E9C54040 40404040 40404040 40404040
11CB8    40404040 40404040 40404040 40404040 40404040 40404040 4040F4F0 F5F64040                ....0        4056
11CC0    40404040 40404040 40404040 40007906 04F04040 40404040 40404040 40404040   06 00D8E    73100003 A
11CC8    40F0F640 F0F0C4F8 C5404040 40F7F3F1 F0F0F0F0 F340C140 40404040 40404040     MTB,1     R3
11CD0    404040D4 E3C26BF1 40404040 D9F34040 40404040 40404040 40404040 40404040
11CE8*   40404040 40404040 40404040 40404040 40404040 D9D9F6F6 40F6F6F6 F6F60000     JJ    RRRRRRRRRRRRR66 66666..
11CF0    404040D1 D1404040 40D9D9D9 D9D9D9D9 D9D9D9D9 D9D9F6F6 40F6F6F6 F6F60000  ..........B5...........2 ......
11CF8    40000008 25000111 200FFFF8 85000004 4B00E065 32100C00 8240000C 00070E58

1BB00    00000000 00790601 40404040 4040404C 40F2F1F7 F14BF361 C2C5E940 40404040  ........            2171.3/BEZ
1BB08    4040F2F2 F3F760D9 C5E24040 40404040 4040404C 40404040 40404040 40404040    2237 RES
1BB18*   40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040
1BB20    40007906 04404040 40404040 D5D6C4C5 D3404040 40404040 4040404C 40404040  ....          NODEL
1BB38*   40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040
1BB40    40007906 04404040 40404040 40404040 4040404C F4F9F461 C2C54040 40404040  ....                494/BE
1BB48    404040F4 F9F661C2 C5404040 40404040 4040F5F0 F561C2C5 E9404040 40404040    496/BE          505/BEZ
1BB50    40F5F0F7 60D9C5E2 40404040 40404040 40404040 40404040 4040404C 40404040  507 RES
1BB58    40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040
1BB60    40007906 04404040 40404040 D5D6C4D9 C3404040 40404040 40404040 40404040  ....          NODRC
1BB78*   40404040 40404040 40404040 40404040 40404040 40404040 40404040 40404040
1BB80    40007906 04404040 40404040 40404040 404040F1 F6F1F060 D3E64040 40404040  ....                1610 LW
1BB88    4040F1F6 F1F761C2 40404040 40404040 40F1F6F2 F261C240 40404040 40404040    1617/B          1622/B
1BB90    F1F6F2F7 61C24040 40404040 404040F1 F6F3F261 C2404040 40404040 4040F1F6  1627/B          1632/B          16
```

Figure 63. CPOOLs and Corresponding SPOOLs Display

```
MONITOR JIT:

*** PHYSICAL PAGE#46   ( 8C00 )

8C00    00000000 00000000 00000000 00000000 00000000 00000000 00000000 0003603B   .............................. .
8C08    00000000 00000000 00000000 00000000 00000000 00000000 00000000 00068939   ................................
8C10    00000000 80700000 00000000 00000000 00000000 00000000 00000000 00000000   ................................
8C18    0000C840 00000C00 CFFE0000 00000000 00000000 00000000 00000000 00000C00   ... ............................
8C20    0000C000 00000000 C2000000 00000000 00000000 00000000 00000000 00000000   ................................
8C28    00000000 00000000 00000000 00000000 0000000C 00000003 0001E4C3 00000C00   ..........................JC....
8C38*   00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   ................................
8C40    00000000 00000000 00000000 00000000 0000000C 00000000 0000000F 36314800   ................................
8C48    00000000 00000000 00008041 003E00F6 00C0C163 00000000 00007153 0000C088   ................6..A............
8C50    0000C008 00060000 00000024 00002FFD 00000044 00000000 6A7006E5 00002808   .......................... ..V....
8C58    0000C001 0001BA80 0000C85C 0000D172 00000082 0000D01A 0001BA99 00005A19   ............H*..J....2......... .
8C60    3000597E 10000000 00000000 000010E8 FF000012 F0000008 00000003 80000000   ...*............Y....O..........
8C68    00000000 00000000 00000007 00000886 00000080 00000080 00004A90 000010E8   ......................... ....Y
8C70    10000000 0000000C 0001BA99 00005A19 3000597E 10000000 000000FF 000010E8   ................ ..............Y
8C78    FF00C012 F0000008 00000003 80000000 00000000 00000000 00000007 00000886   ....O...........................
8C80    0000C080 00000080 33002B99 000010E8 1800000C 00000000 0001BA99 00005A19   ................Y............. .
8C88    3000597E 10000000 00000089 000010E8 FF000012 F0000008 00000003 80000C00   ...*............Y....O..........
8C90    00000000 00000000 00000000 00000886 00000080 00000080 00004A90 000010E8   ......................... ....Y
8C98    10000000 00000000 0001BA99 00005A19 3000597E 10000000 000000FF 000010E8   ................ ..............Y
8CA0    FF00C012 F0000008 00000003 80000000 00000000 00000000 00000000 00000886   ....O...........................
8CA8    00000080 00000080 33002B99 000010E8 18000000 00000000 0001BA99 00005A19   ................Y............. .
8CB0    3000597E 10000000 00207000 000010E8 FF000012 F0000008 00000003 80000000   ...*............Y....O..........
8CB8    0000C000 00000000 C0000000 00000886 00000080 00000080 00004A90 000010E8   ......................... ....Y
8CC0    1000C000 00000000 0001BA99 00005A19 3000597E 10000000 000000FF 000010E8   ................ ..............Y
8CC8    FF00C012 F0000008 00000003 80000000 C0000000 00000000 00000000 00000886   ....O...........................
8CD0    00000080 00000080 33002B99 000010E8 18000000 00000000 0001BA99 00005A19   ................Y............. .
8CD8    2C002BD6 10000000 0001BA99 000000FF FF000012 00000008 00000003 80000000   ...0............................
8CE0    00000000 00001A38 000010E8 00000001 0A000000 00002BB7 000000E1 00000001   ..............Y.................
8CE8    1800C000 FFFF8783 0001BA99 1800000C 30005926 10000000 20002BD6 00007E53   ...........................0....
8CF0    00000006 00000003 00000000 FFFFFFFF 0000000C FFFFFFFF 00000010 00000000   ................................
8CF8    FFFFFFFF 00007E66 00000001 18000000 00000004 0001BA99 00000012 00005B66   ..............................$.
8D00    3000597E 10000000 00207000 000010E8 FF000012 F0000008 00000003 80000000   ...*...............Y....O.......
8D08    00000000 00000000 00000000 00000887 00000080 00000080 00004A90 000010E8   ......................... ....Y
8D10    10000000 00000000 0001BA99 00005A19 000C5BA7 FFFF8785 0001BA99 00002BF2   ................ ...$..........2
8D18    FF00C012 00000008 00000003 0000000C 000C5A52 00000000 0000000C 00005AFF   ...................... ........ .
8D20    00001032 00000003 03007C04 0000000C 0001BC00 00000012 00000000 00007C93   ...........@...................@.
8D28    000011D8 000240F2 08000000 000000C9 000C11D8 0006EC00 0018AB81 44444400   ...W.. 2........I...Q...........
8D30    000011D8 10000000 0000566C 00000043 00000002 00000000 00000057 00000000   ...W....... .....................
8D38    0001BC00 00000012 00000000 00007C93 000011D8 00003E5D 0000000C 44444400   .................@....Q...).......
8D40    00000800 00001324 FFFFFFFF FFFFFFFF 7FFFFFFF FFFFFFFF 00000000 00000000   ................. ..............
8D48    00000000 00000000 00000000 00000000 00065000 00000000 00000000 00000000   ................... ............
8DE8*   0000C000 00000000 00000000 00000000 00000000 0000000C 00000000 00000000   ................................
8DF0    00000000 00000000 00000000 0100001D 00000000 00000000 000056D9 00000000   ...........................K....
8DF8    00000000 00000000 00000000 00000000 00000000 00000000 00000000 03007C04   ..............................@.
```

Figure 64. Monitor JIT Display

```
CURRENT USER:

 USER# 3C

*** PHYSICAL PAGE#CA   ( 19400 )

08C00  0C000282  7AC2C3C8  E2C7C5D5  E2D7D9F0   |....:BCHSGENSPRO|   |     DD          |   |J:JIT    |J:ACCN   |         |J:UNAME |
08C04  FCFCF0F0  F6F2F9F6  C000000F  00000004   |00006296........|   |LCF PACK         |   |         |         |J:CTIME  |J:OVHTI<|
08C08  0C00140D  000364AC  00001AF0  00A5778E   |................|   |                 |   |J:CALCN< |J:PTIME  |         |        |
08C0C  00000017  00000038  00000608  00000034   |................|   |                 |   |J:UTIME  |         |         |J:DELTA<|
08C10  0C25B2ED  80700000  00200009  001A0000   |..2.............|   |                 |   |J:MRT    |J:ABC    |J:RNST   |        |
08C14  80000200  00040000  4E1E0000  00060000   |................|   |           TDV   |   |J:ASSIG< |         |         |J:CPPB  |
08C18  4E1E0140  00100000  4E1E0000  00000000   |+... ....+.......|   |TDV        TDV   |   |J:TRAP   |         |J:JIP    |J:INTER |
08C1C  4E1E0000  00010000  00000000  00000000   |+...............|   |TDV              |   |         |JB:STEP  |J:ASPIN  |        |
08C20  0C000000  00002002  02000000  00000000   |................|   |           NOP   |   |         |J:TELFL< |J:CASSI< |J:INTEN<|
08C24  0C000000  FFFF9600  0040C164  0000C0B8   |.......... A.....|   |                 |   |J:TIMEN< |J:UTIME< |J:USENT  |J:TCB   |
08C28  0C00C200  0000A200  00001368  00008C00   |..B.............|   |                 |   |J:TREE   |J:IPOOL  |         |J:USCDX |
08C2C  00009000  00000303  0001E4C3  00000000   |...........UC....|   |                 |   |J:DCBLI< |M:UC     |J:TITLE  |        |
08C30  0C00270F  0000270F  0000270F  0000270F   |................|   |                 |   |         |         |         |        |
08C34  00001388  00001388  0000001F  0000001F   |................|   |                 |   |         |         |         |        |
08C38  00001388  00002710  00000000  00000000   |................|   |                 |   |         |         |         |        |
08C3C  0C000000  00000000  00000000  00000000   |................|   |                 |   |         |         |         |        |
08C40  00000000  000A0000  00000000  12C012C0   |................|   |              LD |   |         |         |JH:PC    |J:XP    |
08C44  800906C0  00000000  00003C0F  36314800   |................|   |              DW |   |JB:PRIV  |J:ABUF   |J:PUF    |JB:PCW  |
08C48  0C000000  00001388  00008C48  007A0000   |..............:..|   |                 |   |J:IDELI< |         |TSTACK   |        |
08C4C  2CCCC1A3  00000000  00007153  0000C088   |..A.............|   |AI               |   |         |         |         |        |
08C50  0C00006E  00060000  00000024  0000C017   |................|   |                 |   |         |         |         |        |
08C54  0000C38C  00000078  68300600  00002808   |..C.............|   |           BE    |   |         |         |         |        |
08C58  0C0000C1  0001BA80  0000C12D  0000D172   |..........A...J.|   |                 |   |         |         |         |        |
08C5C  0C00D082  0000D01A  0001BA99  00008C00   |...2............|   |                 |   |         |         |         |        |
08C60  0C00907C  0000C3F0  00000000  00000001   |....$..CC........|   |                 |   |         |         |         |        |
08C64  00000000  00005230  00008C65  000064D4   |...............M|   |                 |   |         |         |         |        |
08C68  0C0C3874  000071A6  00000028  00000001   |................|   |                 |   |         |         |         |        |
08C6C  0C007087  00FFFFFF  0000907C  00007068   |............$...|   |                 |   |         |         |         |        |
```

Figure 65. Current User's JIT Display

```
CONTENTS OF TSTACK:

ADDRS     STACK OFFSET    CONTENTS    RELATIVE LOC

 8C4A      0              00008C86    J:CPROC<+.40
 8C4B      1              003F003B
 8C4C      2              27C0C6BE    LOGON-.2B
 8C4D      3              00000000
 8C4E      4              04900001
 8C4F      5              00000000
 8C50      6              00000000
 8C51      7              0000C62B    LOGON+.68
 8C52      8              0000000E
 8C53      9              40404040    T:RCE+.37
 8C54      A              00000039
 8C55      B              00000007
 8C56      C              D6D54040
 8C57      D              0049014E
 8C58      E              10390A00
 8C59      F              0000CD31    LOGON-.69E
 8C5A     10              0000C6AB    LOGON-.15
 8C5B     11              F5F74005
 8C5C     12              D6000000
 8C5D     13              40404040    T:RCE+.37
 8C5E     14              04900001
 8C5F     15              00008C00    J:UN
 8C60     16              00008C2D    M:UC
 8C61     17              0000C865    LOGON-.1D2
 8C62     18              00000010
 8C63     19              10390A00
 8C64     1A              0000CD31    LOGON-.69E
 8C65     1B              00005491    ALTCP+.F         B      ---->  CCORST
 8C66     1C              00008C65    J:CPROC<+.61
 8C67     1D              00008C00    J:UN
 8C68     1E              00008C2D    M:UC
 8C69     1F              FFF8C86B
 8C6A     20              00008C2D    M:UC
 8C6B     21              10390A00
 8C6C     22              0000CD31    LOGON-.69E
 8C6D     23              00006701    MSREXIT+.7       BAL,B  ---->  IOQUEUE<
 8C6E     24              000035F5    T:COCHC+.1       BAL,B  ---->  COCSEND<+.1CC
 8C6F     25              00008C2D    M:UC
 8C70     26              00030807
 8C71     27              00000000
 8C72     28              00000096    SB:FPL+.3
 8C73     29              00002104    MX:PPUT+.1E6     BAL,F  ---->  COCOP+.3D
 8C74     2A              004032AE    COCSEND<+.1EC    BAL,B  ---->  T:REG
 8C75     2B              17000000
 8C76     2C              00005000    T:GL+.22
 8C77     2D              00000000
 8C78     2E              00000011
 8C79     2F              0000003A
 8C7A     30              00008000
 8C7B     31              00000000
 8C7C     32              0000007C    S:MAPCW
 8C7D     33              00000001
 8C7E     34              00030807
 8C7F     35              00000048    S:SIR+.1C
 8C80     36              0000309B    ECHOCR2+.8F      BAL,9  ---->  COCIO+.79
 8C81     37              01003B2E
 8C82     38              000032AE    COCSEND<+.1EC    BAL,B  ---->  T:REG
 8C83     39              0000336B    COCSEND<+.12F    BAL,C  ---->  COCSEND<+.1F1
 8C84     3A              000032AD    COCSEND<+.1ED    BAL,D  ---->  COCENAB<
 8C85     3B              0000A000    LOGON+.2693
 8C86     3C              00005000    T:GL+.22
 8C87     3D              0000588B    T:INITJ<+.9      BAL,B  ---->  T:REG
 8C88     3E              00020000
 8C89     3F              001E0000
 8C8A     40              00000000
 8C8B     41              000010A6    UB:OV
```

Figure 66. Current User's TSTACK Display

Table 51. TSTACK Headings

Heading	Meaning
ADDRS	Virtual address of displayed contents.
STACK OFFSET	Index into stack.
CONTENTS	Contents of stack.
RELATIVE LOC	Address that stack contents point to, in symbol plus displacement form. If the stack cell contains a relative location, the instruction at that location will be displayed if it is an address modifying instruction (e.g., B, BAL, LPSD).

```
ADDITIONAL JIT FOR USER# 3C

*** PHYSICAL PAGE#81    ( 10200 )

8E00     030656CC  2E000002  02049800  2C000800  02029800  2C000800  02030800  2C000800   ................................
8E08     02050800  2C000800  030656CE  2E000002  02049000  2C000800  02046000  2C000800   ..Q........................ ....
8E10     02028000  2C000800  0207E000  2C000800  030656D0  2E000002  02076800  2C000800   ................................
8E18     0207F000  2C000800  02062000  2C000800  02040000  2C000800  030656D2  2E000002   ................................
8E20     02028800  2C000800  0202D800  2C000800  02031000  2C000800  0206E000  2C000800   ..0......................K......
8E28     030656D4  2E000002  02077800  2C000800  02029000  2C000800  02028000  2C000800   ..........J.....................
8E30     02036800  2C000800  030656D6  2E000002  02060800  2C000800  02046800  2C000800   ...M............................
8E38     02032000  2C000800  0202C000  2C000800  030656D8  2E000002  02039000  2C000800   ..........0......@..............
8E40     02035800  2C000800  0206C800  2C000800  02061800  2C000800  030656DA  2E000002   ..................@.............
8E48     02022800  2C000800  02059800  2C000800  02040800  2C000800  02079000  2C000800   ..........H.....................
8E50     030656DC  2E000002  02076000  2C000800  02043000  2C000800  02055000  2C000800   ..................@.............
8E58     02063000  2C000800  030656DE  2E000002  02030000  2C000800  0202E800  2C000800   ......... ................ .....
8E60     02033000  2C000800  02054000  2C000800  030656E0  2E000002  0207C000  2C000800   ...........................Y....
8E68     02032800  2C000800  02027000  2C000800  02058800  2C000800  030656E2  2E000002   .......... .....................
8E70     02036000  2C000800  02075800  2C000800  0202C800  2C000800  02054800  2C000800   ...........................S....
8E78     030656E4  2E000002  02062800  2C000800  02053000  2C000800  02048800  2C000800   .. ...............H.............
8E80     02067800  2C000800  030656E6  2E000002  02025800  2C000800  02044800  2C000800   ...U............................
8E88     02024800  2C000800  02060000  2C000800  030656E8  2E000002  02052800  2C000800   ..........M.....................
8E90     02051000  2C000800  0206F800  2C000800  02064800  2C000800  0306EEEA  2E000002   ...................Y............
8E98     02000000  2C000800  02000000  2C000800  02000000  2C000800  02000000  2C000800   ..........8.....................
8EA0     0306EEEC  2E000002  02000000  2C000800  02000000  2C000800  02000000  2C000800   ................................
8EA8     02000000  2C000800  0306EEEE  2E000002  02000000  2C000800  02000000  2C000800   ................................
8EB0     02000000  2C000800  02000000  2C000800  0306EEF0  2E000002  02000000  2C000800   ................................
8EB8     02000000  2C000800  02000000  2C000800  02000000  2C000800  0306EEF2  2E000002   ...................0............
8EC0     02000000  2C000800  02000000  2C000800  02000000  2C000800  02000000  2C000800   ...........................2....
8EC8     0306EEF4  2E000002  02000000  2C000800  02000000  2C000800  02000000  2C000800   ................................
8ED0     02000000  2C000800  0306EEF6  2E000002  02000000  2C000800  02000000  2C000800   ...4............................
8ED8     02000000  2C000800  02000000  2C000800  0306EEF8  2E000002  02000000  2C000800   ..........6.....................
8EE0     02000000  2C000800  02000000  2C000800  02000000  2C000800  0306EEFA  2E000002   ...................8............
8EE8     02000000  2C000800  02000000  2C000800  02000000  2C000800  02000000  2C000800   ................................
8EF0     0306EEFC  2E000002  02000000  2C000800  02000000  2C000800  02000000  2C000800   ................................
```

Figure 67. Current User's AJIT Display

```
CONTEXT AREA FOR USER# 3C

USER#    =>03C
ACCOUNT =>:BCHSGEN
ORIGIN  =>BATCH
      DCB NAME  ADDRS   CURRENT STATUS            ASSIGNMENT             CURRENT STATE
      --------  -----   ---------------           ----------             -------------

      M:SGLD    09018   CLOSED - NEVER OPEN       DEVICE                 INACTIVE

9018     33000003  0007D4C5  0A000000  01180000  0000000C  80000011  0000902E  00000000  ......ME........................
9020     00000000  00000000  00009043  00000000  00000000  00000000  00000000  00000000  ................................
9028     0C000000  00000000  00000000  00000000  00000000  00000000  01000003  00000000  ................................
9030     0000C000  00000000  02000002  00000000  00000000  03000002  00000000  00000000  ................................
9038     04000002  00000000  00000000  07000003  00000000  00000000  00000000  08010003  ................................
9040     00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000  ................................
9048     00000000  00000000  00000000                                                    ............
      M:LL      0904C   CLOSED - NEVER OPEN       DEVICE                 INACTIVE

904C     30000103  00040004  00000000  00000000  00000000  00000000  00009062  00000000  ................................
9054     00000000  002C0000  00009074  00000000  00000000  00000000  00000000  00000000  ................................
905C     00000000  00000000  00000000  00000000  00000000  00000000  01000003  00000000  ................................
9064     00000000  00000000  02000002  00000000  00000000  03000002  00000000  00000000  ................................
906C     07000003  00000000  00000000  00000000  08010003  00000000  00000000  00000000  ................................
9074     21100080  6811C876  21100002  6831C891  32300001  2530007E  4811C2EE  2130000C  ......H.......H...........B.....
```

Figure 68. Current User's Context Area Display

```
*** PHYSICAL PAGE#93    ( 12600 )

9000     00009016  06D47AE2  C7D3C400  00009018  04D47AD3  D300000D  0000904C  04D47AC5  .....M:SGLD......M:LL......<.M:E
9008     C6832002  0000907C  04D47AD3  D4000625  000090A8  05D47AC4  C9C30016  000090F8  F......@.M:LM.........M:DIC.....8
9010     05D47AD3  C9C20625  0000911C  05D47AD4  E2C7001A  00009140  00000000  3221C022  .M:LIB.......M:MSG..... ........
9018     33000003  0007D4C5  0A000000  01180000  00000000  80000011  0000902E  00000000  ......ME........................
9020     00000000  00000000  00009043  00000000  00000000  00000000  00000000  00000000  ................................
9028     0000C000  00000000  00000000  00000000  00000000  00000000  01000003  00000000  ................................
9030     00000000  00000000  02000002  00000000  00000000  03000002  00000000  00000000  ................................
9038     04000002  00000000  00000000  07000003  00000000  00000000  00000000  08010003  ................................
9040     00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000  ................................
9048     00000000  00000000  00000000  75200003  30000103  00040004  00000000  00000000  ................................
9050     00000000  00000000  00009062  00000000  00000000  002C0000  00009074  00000000  ................................
9058     00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000  ................................
9060     00000000  00000000  01000003  00000000  00000000  00000000  02000002  00000000  ................................
9068     00000000  03000002  00000000  00000000  07000003  00000000  00000000  00000000  ................................
9070     08010003  0000000C  00000000  00000000  21100080  6811C876  21100002  6831C891  ......................H.......H.
9078     32300001  2530007E  4811C2EE  2130000C  2C745001  000213A9  0800C017  0000C1A9  ..........B....... ...........A.
9080     00D8C487  00000B11  00D89092  0000C032  0008154A  006EA400  0000909F  00000000  .QD......Q.......... .>.........
9088     03800000  0000006C  0000005B  000B154A  04100000  04830004  00000000  00000000  ....... ...$....................
9090     005C0000  000B154A  01000303  08C8C7D7  D9C5C3D6  D5404040  02000202  7AC4F0F0  .*..... .....HGPRECON   ....:D00
9098     C2D64040  03000002  00000000  00000000  07010002  00000000  00000000  00000000  BO  ............................
90A0     D9C5C3D6  D5D64040  B591C01C  C7100006  00000007  00000001  00000000  201FFFFF  RECONO  5...G...................
90A8     50200001  00101324  0A000000  0000C48A  0000C48A  80000B22  000090BE  00000000   .............D...D.............
90B0     00000000  002C0000  000090F0  00000000  0F800001  00000000  00000000  00000000  ...........0....................
90B8     04000000  10000000  00000000  00FE0366  00000000  00000000  01000303  06C7C8D6  .............................GHO
```

Figure 69. Current User's Physical Pages Display

```
MONITOR ROOT:

00000  00000000

00010  023C0000  00000000  00000000  00000000   |................|   |                |   |          |          |          |          |

00014  00000000

00020  F0000000  18000000  020000A8  0E000058   |O...............|   |LCF  SD     LPSD|   |          |          |          |          |
00024  00000011  000001F0  32000024  CC000025   |.......O........|   |        LW  SIO |   |          |          |          |          |
00028  CD000025  69C00028  00000000  70000041   |................|   |TIO BCS     LCF |   |          |          |          |          |
0002C  0F00282C  FFFF0000  2200001C  CC000025   |................|   |XPSD    LI  SIO |   |          |          |ATITLE    |          |
00030  CD000025  22003A98  64000032  69C00030   |................|   |TIO LI  BDR BCS |   |          |          |          |          |
00034  E800005F  01820200  01C20000  02300000   |Y.. .....B......|   |B           LCFI|   |          |ROOTSA    |          |          |
00038  030000D4  2E000004  02000168  2E000000   |...M............|   |     WAIT   WAIT|   |          |          |          |          |
0003C  080002A1  00000000  00000000  00000000   |................|   |PLW             |   |          |          |          |          |
00040  0FA0049A  0FA004AE  0FA0049E  0FA004A2   |................|   |XPSDXPSDXPSDXPSD|   |          |          |          |          |
00044  0FA004A6  0FA004AA  0FA004C4  0FA004C4   |............D...D|  |XPSDXPSDXPSDXPSD|   |          |          |          |          |
00048  0FA004B2  0FA004B6  0FA004BA  0FA004BE   |...2............|   |XPSDXPSDXPSDXPSD|   |          |          |          |          |
0004C  0FA02104  0FA02110  0F8004E4  FF008C00   |.............U....|  |XPSDXPSDXPSDDST |   |          |          |          |          |
00050  0FA004FA  0FA004F2  33000000  33F00066   |.......2.....O..|   |XPSDXPSDMTW MTW |   |          |          |          |          |
00054  33F002C1  33108C07  0FA004E0  0FA02114   |.O.A............|   |MTW MTW XPSDXPSD|   |          |          |          |          |
00058  0FA004D0  0FA004D4  0FA004D8  0FA004DC   |.......M...Q....|   |XPSDXPSDXPSDXPSD|   |          |          |          |          |
0005C  0F800492  0F800496  00000000  000079F2   |...............2|   |XPSDXPSD        |   |          |          |          |          |
00060  0F8C039E  0F800630  0F800628  0F800636   |................|   |XPSDXPSDXPSDXPSD|   |          |          |          |          |
00064  00002C80  00000001  FFFFFFFF  0000003C   |................|   |                |   |MONORG    |S:HIR     |M:RCLOK<  |S:CUN     |
00068  00000008  00000190  00000012  00000002   |................|   |                |   |S:CUPS    |S:BOUAN   |S:CUIS    |S:OUIS    |
0006C  0000000A  00000006  00000003  00000000   |................|   |                |   |S:OUIS    |S:GUIS    |S:BFIS    |S:SIP     |
00070  0003D43C  00000229  00000294  0000000F   |..M.............|   |                |   |S:EVF     |S:IDLF    |S:USID    |S:OPC     |
00074  0000000F  00000000  00003F11  10000000   |................|   |             AD |   |SL:OPC    |          |SSPSD     |          |
00078  00003F11  10000000  00000000  00008C89   |................|   |    AD          |   |          |          |PULLE1    |          |
0007C  00000183  30008000  0000013A  0C008000   |................|   |    AW          |   |S:MAPCW   |          |S:ACCW    |          |
00080  00000082  02008000  55530000  00000000   |................|   |         STH    |   |ONLN      |          |          |          |
00084  000C0000  00E9FD9F  00000000  00000000   |.....Z..........|   |                |   |S:EAF     |SX:HPP    |          |          |
```

Figure 70. Monitor Root Display

```
RBBAT RECOVERY FILE:

*GHOST COMMUNICATION BUFFERS --
0         1307C010  04000604  011783E2  07000282  00060007  01070E5A  12000000  00020607   ...........S............ ........
8         01145484  07000261  01000604  010708C0  07000263  07030607  01070C8C  07000257   ......./........................
10        0A00001A  00000000  00000000  16000000  00000000  00000000  19000000  00000000   ................................
18        00000000  1C000000  00000000  00000000  1F000000  00000000  00000000  22000000   ................................
20        0C000000  00000000  25000000  00000000  00000000  28000000  00000000  00000000   ................................
28        2B000000  00000000  00000000  2E000000  00000000  00000000  3100000C  00000000   ................................
30        0C000000  34000000  00000000  00000000  37000000  00000000  00000000  3A000000   ................................
38        00000000  00000000  00000000  00000000  00000000                                 ....................
*RBBAT ENVIRONMENT --
0         2040CA38  17000000  00000673  000000FF  00000007  00000016  0000000C  00000000   . ........0#......................
8         0C000012  00000000  0000D67A  0000D678  0114025E  0000CA38  00000400  0000004D   ..........0:..0#...j..........(
10        00000022  00000000                                                               ........
*RBBAT STATIC DATA --
0         C000064E  000000A2  00002598  00000000  00000063  00000000  00000050  00000050   ...+......................  ...
8         00000000  00038080  15000000  00000000  00000000  00170294  0000000A  0000002E   ................................
10        00000000  FFFFFFFF  00000000  00000C20  00000000  0000E000  0000D452  02080003   ................................
18        01018001  00080044  00040000  0000800E  00080009  00300000  00018062  00080025   ..........................M.....
20        00078001  00000000  00000000  00000000  0000C488  00000002  00000000  00000000   ................................
28        00000000  00000000  2040CA38  17000000  0000D678  000000FF  00000007  00000016   ..................D.............
30        00000057  00000000  00000012  00000000  0000D67A  0000D678  00000000  0000CA38   ........ ........0#.............
38        0C000400  0000004F  00000000  00000021  0000C03E  00370000  00028008  00000691   ..................0:..0#........
40        00000012  0014025E  0000CA38  00000246  00000000  00000024  00000000  0000CC48   .......|.......................0.
48        00000022  0000006B  00000006  00000000  0000004B  00030CD8  00000000  0000D542   .......j........................
50        0003CEF0  0000D715  01018006  00080005  05060012  0101801E  010181DC  03280015   .......j.................Q......N.
58        0C078000  00028001  010181DC  03060004  010181D8  02080006  01018001  00080008   ...O..P.........................
60        01018090  0008000A  00078000  00028001  010181DC  03060004  010181D8  02080006   ................................
68        01018001  00080008  0101809A  0008000A  0007800A  00028006  00078000  00028001   ................................
70        010181DC  03060004  010181D8  02080006  01018001  00080008  00000002  00000000   ................................
78        00000000  00000000  00000000  00000000  0000DA83  0000DA92  00000020  00000000   ................................
80        00000000  00000017  10002589  00000000  0000DD1A  0000E000  00000004  00000000   ................................
88        0C000000  0000C03C  00000000  0000D678  0000D691  0000000E  0000004D  00000004   ...............0#..0.........(....
90        00000002  0000000C  00000014  00002598  0000D67A  0000D678  0114025E  0000D3FB   .................0;..0#...j...L.
```

Figure 71. RBBAT Recovery File Display

```
USER IDENTIFICATIØN:

USER#      ØRIGIN      ACCØUNT#      USER NAME

 2         GHØST       :SYS
 3         GHØST       :SYS
 4         GHØST       :SYS
 6         GHØST       :SYS
 C         ØNLINE      X             UTS000003171
 19        ØNLINE      :SYS          LBE
 1E        ØNLINE      015112        SHA143450151
 22        ØNLINE      0153JJH       HAR123680153
 2E        ØNLINE      XSTDA02       ØSI000000184
 30        ØNLINE      DIAG09        CLA130803571
 37        ØNLINE      DIAG06        RES072303571
 3C        BATCH       :BCHSGEN      SPR000006296
 3F        ØNLINE      F653C3AR      RAM120483171
 44        ØNLINE      :TAR:DCR      FRA095677721
 51        BATCH       :AOOSGEN      CPV000003171
 56        ØNLINE      7162          SAN000003322
```

Figure 72. User Identification Display

```
PATCH FILE:

/DCT1+.8/            %00,8  PMDAT+.1E,2    .008400E0/
/DCT1+.9/            LCF1,13  ACTSUM+.D2   .00E100E2/
/DCT1+.A/            LCF1,13  .100E0,1     .00E30000/
/DCT1+.C/            %00,13  .10010,1      .D10010/         DISC  A
1/.AEE/               .D0E0                .D0E0/           ***PATCH ØUT 7242'S IN HGP
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ***  START ABSØLUTE SYMBØLS  **********************
#BL:ØFS=.20DE                       SMK A00 1 ØF 3 8/13/73            #11190
#PB:REP=.B9A                              .A00 PLS 28AUG73 01-16 #11232
#SV:LS=9              SV:LSIZP            A00 JED  9/4/73/ 1-37  #11835
#DISKAB=.2CC4 = DISKAB IN M:MØN      A00 ØNLY  A00.LIP.1 ØF 4     #11148
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<< ***  END ABSØLUTE SYMBØLS  ***********************
/DPAK+.57/           ØR,11  .A            B @(ØR,11 10)/               1 ØF / LIP 9/25/73       #1115
/@/                   .0                  CI,11 .8000/                     2 ØF 7 LIP 9/25/73
/@/                   .0                  BCR,4 $+3/                       3 ØF 7 LIP 9/25/73
/@/                   .0                  ØR,11 TABLES+.1C/                4 ØF 7 LIP 9/25/73
/@/                   .0                  EØR,11 TABLES+.13/               5 ØF 7 LIP 9/25/73
/@/                   .0                  ØR,11 10/                        6 ØF 7 LIP 9/25/73
/@/                   .0                  B DPAK+.58/                      7 ØF 7 LIP 9/25/73
/SS+.1DB/            LH,15  FLG,4         B @ (LH,15 FLG,4)/  .A00  ARR  8/18/73 1ØF7       #118
/@/                   .0                  WD,0   .37/    .A00  ARR  8/18/73  2ØF7
/@/                   .0                  LH,15  FLG,4/   .A00   ARR 8/18/73  3FØ7
/@/                   .0                  EXU SS+.1DC/   .A00  ARR 8/18/73  4ØF7
/@/                   .0                  STH,15  FLG,4/    .A00  ARR 8/18/73  5ØF7
/@/                   .0                  WD,0   .27/   .A00 ARR  8/18/73  6ØF7
/@/                   .0                  B SS+.1DE/   .A00  ARR  8/18/73  7ØF7
/SS+.354/            BNE  SS+.356         BEZ  SS+.367 (BNEZ SS+.356)/ .A00 ARR 8/21/73 1ØF3    #119
/SS+.355/            B  SS+.367           LCF,2   .8C00  (B SS+.367)/ .A00 ARR  8/21/73 2ØF3    #119
/SS+.356/            BCS,4  SS+.361       BCR,12 SS+.361 (BCS,4 SS+.361)/.A00 ARR  8/21/73 3ØF3  #119
/BL:ØFS/              .3B                 ;Q-3/                     SMK A00 2 ØF 3 8/13/73      #11
/ØS+.1FA/            BGE  CUN+.197,4      B ØS+.1FE/                SMK A00 3 ØF 3 8/13/73      #111
/CØØP+.1FD/          BAL,4  SGCQ+.6       BAL,4 @(BAL,4 CØØP+.212)/ A00K RJR 9-14'CIRCSGCHD'01-12 #11191
/@/                   .0                  LH,5    SGCHD/                   A00K RJR 9-14'CIRCSGCHD'02-1
/@/                   .0                  B    CØØP+.212/                  A00K RJR 9-14'CIRCSGCHD'03-1
```

Figure 73. Patch File Display

```
INSWAP USER:

 USER# 50

***THIS USER WAS OUT OF CORE

 08C00  00000004  E7E2F3F0  F2E34040  C2C5C5F0   |...MXSTO2T  BEEO|   |     EXU LB      |   |J:JIT     |J:ACCN    |          |J:UNAME  |
 08C04  F2F2F0F6  F0F1F8F6  00000001  0000001A   |22060186........|   |LB  LCF          |   |          |          |J:CTIME   |J:OVHTI< |
 08C08  0000078A  000056D8  000007AB  00088120   |.......Q........|   |                 |   |J:CALCN<  |J:PTIME   |          |         |
 08C0C  00000000  00000000  00000000  FFFFFF43   |................|   |                 |   |J:UTIME   |          |          |J:DELTA< |
 08C10  00043560  00700000  0C200000  000E002E   |... ............|   |                 |   |J:MRT     |J:ABC     |J:RNST    |J:NRS    |
 08C14  80000200  0000A129  4E1EB129  005C0040   |.......1..1.... |   |           TDV   |   |J:ASSIG<  |          |          |J:CPPO   |
 08C18  4E1E0042  00000000  4E1E0000  00000000   |+.......+.......|   |TDV        TDV   |   |J:TRAP    |          |J:JIP     |J:INTER  |
 08C1C  4E1E0000  01010000  00000000  00000000   |+...............|   |TDV              |   |          |JB:ORG    |J:ASPIN   |         |
 08C20  00000000  00002002  02000000  00000000   |................|   |           NOP   |   |          |J:TELFL<  |J:CASSI<  |J:INTEN< |
 08C24  00000000  FFFFA17D  0C10B14E  0000A000   |........'..1+...|   |                 |   |J:TIMEN<  |J:UTIME<  |J:USENT   |J:TCB    |
 08C28  0000AC00  000003E8  00009C00  0000A800   |........Y.......|   |                 |   |J:TREE    |          |J:USCOX   |J:DCBLI< |
 08C2C  00000003  0001F4C3  000003E7  0000270F   |.......UC...X....|  |                 |   |M:UC      |J:TITLE   |          |         |
 08C30  0000270F  0000270F  0000270F  000001F4   |...............4|   |                 |   |          |          |          |         |
 08C34  000003E8  00000012  000003E8  000003E8   |...Y........Y...Y|  |                 |   |          |          |          |         |
 08C38  00000000  00000000  00000000  00000000   |................|   |                 |   |          |          |          |         |
 08C3C  00000000  00000000  00000000  00000000   |................|   |                 |   |          |          |          |         |
 08C40  000A0000  00000000  12C012C0  40000008   |............ ...|   |           LD  TTBS|   |        |JH:PC     |J:XP      |JB:PRIV  |
 08C44  00000000  0000500F  36314800  FFFFFF06   |...... ........|    |           DW    |   |J:ABUF    |J:PUF     |JB:PCW    |J:IDELT< |
 08C48  00000000  000003E8  00008C83  00420038   |........Y.......|   |                 |   |J:EXTEN<  |          |TSTACK    |         |
 08C4C  20C0AEC0  10000000  00000000  00000000   |................|   |AI   AD          |   |          |          |          |         |
 08C50  9000BE46  00000292  00000000  00000000   |................|   |AD               |   |          |          |          |         |
 08C54  00000001  00000001  00000079  00000000   |................|   |                 |   |          |          |          |         |
 08C58  00000040  01850000  FFFF0000  02008001   |... ............|   |                 |   |          |          |          |         |
 08C5C  00000002  0000ADB9  0000AEC0  00008C00   |................|   |                 |   |          |          |          |         |
 08C60  0000AA00  0000A07E  00000011  00000040   |...........0.... |  |                 |   |          |          |          |         |
 08C64  01850000  00004DF4  00008C65  000072A6   |.......(4.......|   |                 |   |          |          |          |         |
 08C68  20009C40  0600AA00  20009C40  0000AA00   |... ....... ....|   |AI   CAL3AI      |   |          |          |          |         |
 08C6C  FFF8A085  0600AA00  00000040  01850000   |.8......... ....|   |DST  CAL3        |   |          |          |          |         |
 08C70  000072A6  00040000  20406087  17000000   |.........  .....|   |           AI    |   |          |          |          |         |
```

Figure 74. Inswap and Outswap User's Core Display

```
INCORE USERS:

 USER# 1E

*** PHYSICAL PAGE#43   ( 8600 )

 08C00  8000001E  F0F1F5F1  F1F24040  E2C8C1F1   |....015112  SHA1|   |     LCF CB      |   |J:JIT     |J:ACCN    |          |J:UNAME  |
 08C04  F4F3F4F5  F0F1F5F1  0000001E  00000010   |43450151........|   |STCFLCF          |   |          |          |J:CTIME   |J:OVHTI< |
 08C08  00001659  00002A8B  0000140A  00047663   |................|   |                 |   |J:CALCN<  |J:PTIME   |          |         |
 08C0C  0000202A  000006AD  0004998A  FFFFFFA2   |................|   |                 |   |J:UTIME   |          |          |J:DELTA< |
 08C10  00000000  00700000  00200000  00000000   |................|   |                 |   |J:MRT     |J:ABC     |J:RNST    |         |
 08C14  80000000  00000000  4E1ECC0F  00017FFF   |..........+..... .|  |           TDV   |   |J:ASSIG<  |          |          |J:CPPO   |
 08C18  07000000  00000000  07000000  000000D9   |...............R|   |CAL4       CAL4  |   |J:TRAP    |          |J:JIP     |J:INTER  |
 08C1C  07000000  0A000000  00000000  00000000   |................|   |CAL4PLM          |   |          |JB:STEP   |J:ASPIN   |         |
 08C20  00000000  00100054  02000000  00000000   |................|   |           NOP   |   |          |J:TELFL<  |J:CASSI<  |J:INTEN< |
 08C24  00000000  FFFF9A94  00000000  00000000   |................|   |                 |   |J:TIMEN<  |J:UTIME<  |J:USENT   |J:TCB    |
 08C28  0C00C800  0000AC00  00000FA0  00000000   |..H.............|   |                 |   |J:TREE    |J:IPOOL   |          |J:USCDX  |
 08C2C  00009000  00781003  0006903E  0002CDEC   |................|   |                 |   |J:DCBLI<  |M:UC      |J:TITLE   |         |
 08C30  0C000000  00100000  00000000  00100000   |................|   |                 |   |          |          |          |         |
 08C34  0000CDEC  00000000  70F30000  00000000   |.........'3.....|   |           DC    |   |          |          |          |         |
 08C38  00000000  00000000  00000008  00000000   |................|   |                 |   |          |          |          |         |
 08C3C  0A132500  00000000  00000000  00000000   |................|   |PLM              |   |          |          |          |         |
 08C40  00000000  4E1E0000  00090000  12C012C0   |....+...........|   |     TDV      LD |   |          |          |JH:PC     |J:XP     |
 08C44  80090400  00000000  16001E00  361E506E   |.............. >|   |              DW |   |JB:PRIV   |J:ABUF    |J:PUF     |JB:PCW   |
 08C48  FFFFFF06  00000F9A  00008C89  003C003E   |................|   |                 |   |J:IDELT<  |          |TSTACK    |         |
 08C4C  20CCCBF0  00000000  00000000  1110003D   |...0............|   |AI            CD |   |          |          |          |         |
 08C50  00000000  00009286  0000F921  00000003   |..........9.....|   |                 |   |          |          |          |         |
 08C54  0000F923  0000CB99  9400002E  0000F921   |..9...........9.|   |                 |   |          |          |          |         |
 08C58  00000328  C9404040  0001FFFF  00200200   |....I   ........|   |     OR          |   |          |          |          |         |
 08C5C  00000000  00000FBF  0000CA07  00008C00   |................|   |                 |   |          |          |          |         |
 08C60  000C9286  0000C01E  00000010  00000328   |................|   |                 |   |          |          |          |         |
 08C64  C9404040  0000523D  00008C65  00006490   |I    ...........|   |OR               |   |          |          |          |         |
 08C68  000001EC  04007170  00007087  000A09C6   |...............F|   |     CAL1        |   |          |          |          |         |
 08C6C  00000089  0000AA00  0003E670  0000706B   |..........W.....|   |                 |   |          |          |          |         |
```

Figure 75. Incore User's Core Display

```
SYMBOL MAP:

('-' = ABS DEF; '<' = TRUNCATED SYMBOL)

16    -:AMHED    6061  :LBL      7E    -:LOGSZ    0     -:9        288   #SWAP@C<  6290  @SGCBUF   16    •ABA
28ED  ABORT      1     -ACCN     37    -ACCTSIZ   2916  ACCT       12FE  ACNCFU    3F    -ACNTBLM  4FE   ACNTBL
5561  ACTIVAT<   3E4E  ADD1      2672  ADJSTCN<   1     -AIF       2     -AIFJE    3     -AIFNC    448   ALLOBUT
5DB4  ALLOG      5DB8  ALLOQWB   5DAC  ALLOREG    449   ALLYON     19    -ALOCCT   524C  ALTCP     3810  ANS$AVR
1     -ANS       3     -ANSERR#  2007  ANSFLGS    38B1  ANSMOVE    20D9  ANSPRT    4     -ANSREEL< 4     -ABFL
6     -ABFNB     7     -ABFP     5     -ABF       6F2   ARSZ       65DD  ASPIN     497A  ASP1      2E    •ATITLE
20B4  AVRFNMT    2536  AVRID     253D  AVRNOU     23EC  AVRTBL     5FBC  AVR       20D2  AVRSID    D     •AVRTBLN<
5     -AVRTBLS<  5FF5  AVRX      44    -BAABC     3858  BAAVRFN<   465   BALANCE   265   BANPMC    48    •BARNST
C     -BATAPE    9     -BATYC    62F6  BCDMESS    3B80  BCDTOEB<   62F8  BCDWRT    14    -BCSTGFL  7FDA  BEGINOF<
7FE8  BEGINON    6CF8  BFRMWR    F     -BGNPMPR<  58C   BGRAN      1324  BGRCFU    20DD  BL:IFS    20DE  BL:OFS
54F   BLANK      6DB8  BLKIN     796B  BLNKFIL<   2E    BOOTENT    542   BOOTIC    2836  BOOTSBA<  266C  BOTTOM
19    -BTCBIT    3AB   RT31T80   728   BUFCNT     5CCC  BUFF       2674  BUFLAGS   280   BUFLIMS   265A  BUFMASK
594   BUF1       5D6   BUF2      68E2  BUF2X      372D  BUSER1     2108  BUSER10   210C  BUSTEMP   20DC  BW:CHG
2CC   C:CAL      2C3   C:CIT     2C4   C:CITI     2CE   C:CI       2D2   C:COS     2CF   C:CO      2D1   C:CSC
2CD   C:CTW      2C2   C:ETM     2F2   C:IDLES<   2F0   C:IDLES    2EF   C:IDLE    2F1   C:IDLEW   386   C:LAST
2EC   C:MSB      2DF   C:MSO     2C6   C:NOPRO<   2CB   C:NSP      2EE   C:OVHD    2ED   C:PROCN   2C5   C:PROCR<
2C7   C:RECYC<   2D0   C:RTRW    2BD   C:RT90     2BE   C:SC       2E0   C:SCB     2F9   C:SCI     2D3   C:SCO
2F3   C:SIT      2F4   C:SRT     2F5   C:STT      2F7   C:ST       2C0   C:TIC     2C1   C:TINC    54CF  CALBAD
524C  CALCK      517E  CALPR     5C55  CAL1P      4B2   CAL1PSD    5180  CAL1P11   51AA  CAL1IN3   6315  CAL1IN7
486   CAL2PSD    5C35  CAL2XXX   48A   CAL3PSD    5C3F  CAL3XXX    4BE   CAL4PSD   5C49  CAL4XXX   22    •CASSIGN
2666  CATBUF     267D  CBAHD     6923  CBB4       6933  CBB5       267C  CBFHD     10    -CBPOOL   267D  CBRHD
20    •CBSIZE    12    -CCBEF    2     -CCBPOOL<  21    -CCLFLAG<  21    -CCLTFLG< 531A  CCURST    531A  CC1RST
5321  CC1SET     531F  CC2SET    3     -CDBPOOL<  18    -CDPO      6     -CEXT     619   CFLAG     1     •CFUBIT
13    •CFUSIZE   352   CH:DC     32A   CH:DIT     2FA   CH:DI1     302   CH:DI2    30A   CH:DI3    362   CH:DLI
372   CH:DLO     31A   CH:DOS    312   CH:DOT     322   CH:DRT     342   CH:DT     332   CH:DTT    283   CH:SWAP<
29D6  CHGMOD     37CA  CHKANSO   37DQ  CHKANS1    65AE  CHKBIT1    65B0  CHKBIT    65AD  CHKBITO   65F6  CHKCUR
5795  CHKDA      5798  CHKDAG    3E9D  CHKPROT    6491  CHKWAT     261   CHNMSK    13    •CIL      2B4   CITHEN
9     •CITSIZ    23AA  CIT1      23AD  CIT2       2380  CIT3       2383  CIT4      238D  CIT5      236D  CIT6
4F    •CJOB      545C  CKLIMIT   48A   CK3FLG     65D9  CLEARHI    211A  CLIST     38DA  CLKOV     4DC   CLK1PSD
5C33  CLK1XXX    558A  CLK2I     4D4   CLK2PSD    382   CLK2RAN<   385   CLK2TMP   4D8   CLK3PSD   4DC   CLK4PSD
3EF8  CLK4       7E40  CLOCKI    582   CLOCKTM<   6DF3  CLOSEPV    6900  CLRBBUF   68C2  CLRBFUB   6941  CLRB
```

Figure 76. Symbol Map Display

TABLE OF CONTENTS:

Figure 77. Table of Contents for Analyze Dump

ANALYZE MESSAGES

Table 52 contains the messages that are output by Analyze. Most of these messages identify error conditions. Others merely supply information.

ANALYZE COMMAND SUMMARY

Table 53 summarizes Analyze commands. The left-hand column contains the command format, the right-hand column contains the command description.

Table 52. Analyze Messages

Message	Description
ANLZ: ENTER COMMAND, N/L SAYS TO DO ALL	This message issued to operator after GJOB ANLZ key-in. Operator may respond with one of the following: NO = just exit TA = read recovery-built tape HE = run interactively from console CP = read CP5DMP file 0-7 = read indicated MONDMP file N/L = do default ghost run
BAD COMMAND	The command was unrecognizable.
CANNOT OPEN FILE name	The file specified by the INPUT command cannot be opened.
CAN'T GET THE BUFFER	The user was not allowed enough core in his account to read in the monitor symbol stack.
COUNT ERROR TAIL ERROR	The tail and last page in a chain do not agree.
ENTER TAPE TYPE: 9T, BT, etc...	The user must supply the tape type if tape input is to be used.
ERR/ABN CODE = xxxx**dcb	An I/O error or abnormal condition occurred during an INPUT operation. xxxx is the error or abnormal code. dcb is the address of the DCB associated with it.
LOC1 GREATER OR EQUAL LOC2	The first location entered for a loc_1,loc_2 (or similar) command was greater than the second location.
xx PRIVILEGE LEVEL NOT HIGH ENOUGH	The user privilege level was not high enough for the requested operation.
SORRY, NO PAGE xx	The page containing the location specified by the user was not found in the input file.
THE LAST PHYSICAL PAGE IN THE FILE IS xx	The size of the file read from tape by the INPUT command is specified by the last physical page in the file.

Table 53. Analyze Command Summary

Command	Description
↑	Dumps the last location and is used in connection with loc and loc_1, loc_2.
*	Dumps the indirect location and is used in conjunction with loc and loc_1, loc_2.
AL[L]	Performs the functions of the INPUT, DISPLAY, and RUN commands and of Analyze (except dumps) when initiated by the Automatic Recovery Procedure. A numerically and alphanumerically sorted monitor map is output at the end of the ALL display.
BF fid	Specifies the name of the boot file that represents the monitor being examined by Analyze. The file M:MON in :SYS is assumed by default.
CL[OSE]	Causes input dump file to be closed.
CO[MPARE],loc_1,loc_2	Compares the dump (locations loc_1 through loc_2) with the running monitor and outputs the locations with nonequal contents.
DE[LTA]	Associates the debugger Delta with Analyze and gives control to Delta when the BREAK key is hit by the user.
DI[SPLAY] option	Outputs information existing at the time of the crash. The options are AJ[ITS] – JIT, AJIT and context of all incore users. AT[ABLE] – contents of ALLYCAT's tables used to manipulate buffers. AV[R] – contents of AVR tables. CI[TS] – contents of CITs. CO[C] $\left[,\left\{\begin{matrix}S\\id\end{matrix}\right\}\right]$ – contents of COC tables. id specifies user, S indicates all users. CU[N] – contents of user's JIT, AJIT, and context. DC[TS] – contents of DCTs. EL[OG] – incore error log entries. FQ – contents of I/O tables not currently in use. ID $\left[,\left\{\begin{matrix}S\\id\end{matrix}\right\}\right]$ – user's logon identification as it appeared in his JIT. id specifies user, S indicates all users. IO[Q] – contents of CIT, DCT, and IOQ tables. IQ – contents of IOQs. JI[T], id [,loc_1, loc_2] – JIT contents for specified user (id).

Table 53. Analyze Command Summary (cont.)

Command	Description
DI[SPLAY] option (cont.)	MR – contents of monitor root. OJ[ITS] – JIT of all users not in core. PA[RTITION][, {S / id}] – partition table values. id specifies user, S indicates all users. PF[ILE] – contents of patch file. PM – contents of page matrix. PP,pgno – contents of physical page of memory. PR[OCESSOR][, {S / id}] – contents of processor tables. id specifies a user, S indicates all users. RA[T] – contents of resource allocation tables. RE[GISTERS] – contents of registers and cause of crash. RC[XT] – recovery context. ST – contents of output symbiont tables. SW[AP] – contents of swap tables. SY[MBIONTS] – contents of symbiont tables. TR[APS] – contents of trap and interrupt locations. TS[TACK][,[id][, #levels] – dumps out the indicated number of Temp Stack levels (default 21_{10}) for the indicated user (user # =0 for monitor's stack), displaying values as symbol + displacement. If the stack cell contains a monitor address, the instruction at that location will be displayed if it is an address-modifying instruction (e.g., BCS, BAL, or XPSD). US[ER][, {S / id}] – contents of tables for specified user. id specifies a user, S indicates all users. VP, pgno – contents of virtual page of memory.
DU[MP] loc_1,loc_2	Dumps specified range of addresses.
EN[D]	Exits from Analyze.
HE[LP]	Lists all Analyze commands.
IN[PUT] option	Directs Analyze to input from a particular disk or tape file or to open a file. The options are ;[V] – reads a tape created by executive Delta. L[AST] – opens the last file formed by the recovery procedure.

Table 53. Analyze Command Summary (cont.)

Command	Description
IN[PUT] option (cont.)	number – opens the numbered crash file formed by the recovery procedure. T[APE] – reads a labeled tape created by the recovery procedure. C[P5DUMP] – opens the CP5DUMP file.
Line Feed (or carriage return)	Dumps the contents of the next location and is used in conjunction with loc and loc_1,loc_2.
loc	Outputs the contents of the specified location.
loc_1,loc_2	Outputs the contents of memory locations between loc_1 and loc_2.
loc = value	Places the value in the specified location of the running monitor.
LP[rows]	Directs the output of Analyze to the line printer, where rows is dump width in hexadecimal words. Default is full line.
MA[P],id	Loads the map of the specified user if his JIT is in core.
MO[NITOR][DI[SPLAY]]	Turns the monitor display mode on and off. MONITOR turns the display mode off. MONITOR DISPLAY turns the display mode on.
NO[DELTA]	Disassociates the debugger Delta from Analyze.
PR[INT]	Closes the output symbiont file to allow output to the line printer without requiring a return to TEL.
RO[WS], value	Establishes width of dump output in number of words, where value may be 1 through 12.
RU[N] option	Outputs various linked lists of the monitor by running through the list and displaying each entry. The options are MO[NITOR] $\left[,\begin{Bmatrix}\text{S}\\ \text{pgno}\end{Bmatrix}\right]$ – specifies monitor pages. S, the default, indicates all. A specific page may be requested. PR[OCESSOR] $\left[,\begin{Bmatrix}\text{S}\\ \text{name}\end{Bmatrix}\right]$ – specifies processor pages. S, the default, indicates all. A particular processor may be specified. ST[ATE] $\left[,\begin{Bmatrix}\text{S}\\ \text{q\#}\end{Bmatrix}\right]$ – specifies state queues. A particular queue number may be specified, or S, the default, indicates all.

Table 53. Analyze Command Summary (cont.)

Command	Description
RU[N] option (cont.)	US[ER] $\left[,\begin{Bmatrix}S\\id\end{Bmatrix}\right]$ – specifies user pages for a particular user (id), or for all users (S). S is the default.
SE[ARCH] , value,loc_1,loc_2	Searches for and outputs all words between loc_1 and loc_2 that contain the value under the mask.
SM[ASK] ,value	Sets the mask to the specified value.
SY[MBOLS][fid]	Creates a numerically sorted monitor map, using the fid specified or MONSTK.:SYS.
UC[rows]	Directs the output of Analyze to the on-line terminal, where rows is dump width in hexadecimal words. Default is full line.
UN[MAP]	Turns off the mapping mode of operation.

10. SHARED PROCESSOR MAINTENANCE – DRSP

INTRODUCTION

Development and check out of CP-V systems is simplified through use of DRSP (Dynamic Replacement of Shared Processors). DRSP allows replacement, creation, or deletion of shared processors and monitor overlays while the system is operational. The extra processor or overlay space in the shared processor tables must be allocated during system generation (PASS2). Processors that are normally invoked following a recovery cycle (ALLOCAT, GHOST1, RECOVERY, and XDELTA) are not dynamically replaceable.

XDELTA (Executive Delta) is an additional debugging aid that is optionally retained at system initialization. XDELTA is described in the Delta chapter of the CP-V/TS Reference Manual, 90 09 07.

DRSP DESCRIPTION

DRSP can be run either as an on-line or a batch processor. Input can be either from the command device or from a terminal. DRSP is called on-line by entering the name of the processor as a TEL command.

Example:

!DRSP (RET)

DRSP HERE

>

The DCBs used by DRSP which may be assigned by the user are

1. M:SI for command language input.
2. M:LL for terminal output.
3. M:SL for listing of input commands during a batch run and diagnostic message output.

DRSP COMMANDS

The seven DRSP commands are

ENTER

REPLACE

DELETE

LIST

LISTALL

?

END

In the DRSP command descriptions, the term 'proname' refers to the name of a processor or overlay as found in the shared processor tables. The file specified by proname must be in load module format.

All of the above commands except "?" can be followed by comments, which will be printed as part of the command line during a batch run of DRSP. To add comments, terminate the command with a blank character followed by a period. All characters entered after the period are treated as comments. The comments are terminated by (RET) or end-of-card. Comments cannot be continued to the next record.

ENTER The ENTER command is used to enter a new shared processor or monitor overlay into the system.

The format of the command is

E[NTER] proname [{FROM|WITH} fid] [,option] [,option] [,option]

where the options are as follows:

[J][S][D][P][M][T][B][G][C] specifies one or more flags to be associated with the processor. Flags may be associated only with shared processors, not monitor overlays. The flags indicate the following:

J processor is allowed to alter the JIT.

S special shared processor.

D processor is a debugger.

P public library.

M processor allowed maximum memory during execution.

T command processor accessible by terminal users.

B command processor accessible by batch users.

G command processor accessible by ghost users.

C command processor accessible by terminal, batch, and ghost users.

If D or P is specified, S is redundant and is assumed. If the C flag is used, the specific flags (T, B, G) are redundant and should not be used. Various combinations of the above are possible up to a maximum of six characters; e.g., a processor that is allowed to alter the JIT and has maximum memory available for execution would be flagged JM. The flag combination PD or usage of the P flag when the processor name is other than :Pnn results in an error message.

O specifies that proname refers to a monitor overlay and should be entered in the monitor overlay area of the processor tables. If O is not specified, it is assumed that a shared processor is being entered. The flags and O options are mutually exclusive.

PERM specifies that the processor is to be available to users even after a system crash. The processor will be present both in the system account (:SYS) and on swap disk. "Empty" slots must be available in the disk copy of the processor tables. If this option is not used, the new processor version will reside only on swap disk and will be lost in the event of a crash. The version of the processor that will be restored is the version in the system account at the time of the crash.

W specifies that if the proname cannot be entered into the processor table because there are no name slots free, DRSP is to wait until there is a slot available. If this wait option is not specified, the command terminates without entering the new processor. This option is to be used for processors only; it is ignored for monitor overlays.

REPLACE The REPLACE command is used to replace an existing shared processor or monitor overlay. If this command is used, the previous version of the processor is lost. However, current users continue to use the old copy until they are disassociated from the processor.

The format of the command is

R[EPLACE] proname $\left[\begin{Bmatrix}\text{FROM}\\ \text{WITH}\end{Bmatrix}\right]$ fid [,option][,option][,option]

where the options are as follows:

[J][S][D][P][M][T][B][G][C] specifies flags to be associated with the processor. The option is the same as for the ENTER command.

PERM specifies that the new version of the processor is to be available to users even after a system crash. This version of the processor will be present both in the system account (:SYS) and on swap disk. "Empty" slots must be available in the disk copy of the processor tables. If this option is not used, the new processor version will reside only on swap disk and will be overwritten in event of a crash by the processor version in the system account.

W specifies that if the proname cannot be entered into the processor table because there are no name slots free, DRSP is to wait until there is a slot available. If this wait option is not specified, the command terminates without replacing the old processor. This option is to be used for processors only; it is ignored for monitor overlays.

DELETE The DELETE command prevents further user association with a processor. Users associated with the processor when this command is issued will continue to use the processor until they disassociate.

The format of the command is

D[ELETE] proname [, PERM]

where PERM specifies that no new users will ever be associated with this processor (even after a system crash).

LIST The LIST command lists the processor name, the name associated with each entry in the processor name table, and the amount of disk space occupied by the processor.

The format of the command is

L[IST] $\left[\begin{Bmatrix}\text{proname}\\ \#\text{xx[-yy]}\end{Bmatrix}\right]$

where

proname specifies an explicit processor name. (The proname M:DUMLM appears many times in the processor tables. If selected, all these entries will be listed.)

xx-[yy] specifies the name table index or a range of name table indexes to be listed.

Initial use of the LIST command with no proname or index specified will provide a list of each processor table entry and its corresponding table index.

LISTALL The LISTALL command lists each shared processor name and its entries in the following tables:

PB:HPP	Head of the physical page chain
PB:TPP	Tail of the physical page chain
PB:DSZ	Number of data pages
PB:DCBSZ	Number of DCB pages
PH:PDA	Disk address of first procedure page
PH:DDA	Disk address of first page of data and DCBs
PB:UC	Number of users in core using the processor
PB:LNK	Processor number of next overlay
PB:PVA	Virtual page number of first procedure page
PB:HVA	Virtual page number of first unused page
P:SA	Processor flags and start address

The format of the command is

LISTALL $\left[\begin{Bmatrix}\text{proname}\\ \#\text{xx[-yy]}\end{Bmatrix}\right]$

where proname and xx[-yy] are as defined in the LIST command.

? The question mark command requests a detailed error message when an error has been noted by DRSP. The command is applicable only for the on-line mode. Its function is described in detail in the section "DRSP Error Messages". The format of the command is

?

END The END command terminates DRSP. The format of the command is

END

11. When replacing the FILL processor a modified procedure is required: Following REPLACE FILL WITH N.A.P., OPTION 1 thru 3, the user has to abort the FILL ghost. This is done via a message to the operator to key in X, id, where id is the SYSID of the FILL ghost which appears when the message 'REQUEST FILL, NO FILL, OR INSTANT SQUIRREL (F, N, S)' is output on operator's console. This will ensure that the FILL copy in the user swap disk area is destroyed and the replaced version of FILL is brought in the next time FILL wakes up.

DRSP LIMITATIONS AND RESTRICTIONS

The following lists DRSP limitations and restrictions:

1. Only users with a privilege level of C0 or greater are allowed to use the ENTER, REPLACE and DELETE commands. The LIST command requires a privilege level of 80 or greater.

2. There must be sufficient space in the swap disk processor/overlay area to hold the new or replacing entry. This extra space is allocated by SYSGEN PASS2 via a :SPROCS control card.

3. Replaced or entered items must be accessible load modules.

4. Only one level of overlay is permitted in a processor.

5. A monitor overlay must have no tree structure.

6. A monitor overlay is assumed to be DATA only.

7. A processor overlay must be PROCEDURE only.

8. ALLOCAT, GHOST1, RECOVER, XDELTA, M:DUMLM may not be processed with DRSP commands.

9. Overlays for processors cannot be replaced or entered individually.

10. GETs of programs saved with an associated processor most likely will not work if the processor has been changed between SAVE and GET.

DRSP ERROR MESSAGES

The error message structure of DRSP is designed to give a user detailed information when so desired without burdening him with long typeouts when the error is obvious. When running on-line, DRSP will respond to commands in error by typing

EH@ n

where n is the character position at which an error was first detected. If the user requires more information, he responds with a question mark (?). DRSP responds with a detailed error message (see Table 54). If the error is obvious, the user may retype the command (or proceed to the next command). For errors that occur after command syntax is completed, this message changes to

EH

since command character position is meaningless.

In batch mode, the detailed error messages are printed without the interrogative sequence described above.

In addition to error messages, certain other messages are given for information purposes only (see Table 55). No response is expected.

Except where noted, the error condition truncates execution of the requested command.

DRSP COMMAND SUMMARY

Table 56 contains a summary of commands for the DRSP processor. The left-hand column specifies the format and the right-hand column defines the function.

Table 54. DRSP Error Messages

Message	Meaning
BREAK 50 BREAK 51 BREAK 52 BREAK 53	User hit BREAK during DRSP execution. The number defines the point at which the DRSP processor exited, as described in the UTS Reliability and Maintainability Technical Manual, 90 19 90.
CANNOT OPEN THE FID	DRSP cannot access the load module defined by the fid.
CAN'T OPEN M:BO (PERM)	I/O error detected while trying to open the output file in :SYS. The processor is entered/replaced on non-"PERM" basis.
DON'T SET FLAGS WITH MONITOR OVERLAY	Flag setting is not permitted when ENTER or REPLACE monitor overlay is executed.
DON'T USE COMMAND ON TEL/CCI	ENTER or DELETE commands must not specify the proname 'TEL' or 'CCI'.
DRSP I/O ERROR IN READING COMMAND	Error detected in reading DRSP command.
DRSP I/O ERR/ABN (CLOSE)	Error or abnormal condition detected at CLOSE of output file. The processor is entered/replaced on non-"PERM" basis.
DRSP M:BO ERROR (PERM)	I/O error detected while writing or closing the output file in :SYS. The processor is entered/replaced on non-"PERM" basis.
DRSP M:EI ERROR (PERM)	I/O error detected while reading file fid. The processor is entered/replaced on non-"PERM" basis.
DRSP M:EI ERROR (WRITESWAP)	I/O error detected while reading fid for writing on the swap disk.
DRSP PROGRAM ERROR (SHOULDN'T HAPPEN)	DRSP detected contradictory conditions during processing. Requires system programmer intervention.
ERR MSG NOT FOUND. KEY = xxxxxx	No error message corresponds to the error code xxxxxx generated. Please report this system error.
FID IS NOT A LOAD MODULE	Error or abnormal return executed while trying to read the TREE record of the load module specified by fid.
FILE STORAGE LIMIT IN SYSTEM ACCOUNT	When writing the load module into the :SYS account for the PERM option, the file space for that account is exceeded.
ILLEGAL COMMAND	Command entered is not defined in DRSP.
ILLEGAL COMMAND OPTION	An optional parameter typed in the command is not recognized.
ILLEGAL INDEX RANGE	Index specified in LIST/LISTALL command not within legal range of processor name table.
ILLEGAL LMN (LOAD BIAS CHECK)	Illegal load bias detected when processor written to swap disk.
ILLEGAL PRONAME, NOT :PNN FORMAT	A processor flagged as a public library must conform to the name format :Pnn.
ILLEGAL PROTECTION TYPE FOR PUBLIC LIBRARY	The load module for a public library must be root only and procedure only.

Table 54. DRSP Error Messages (cont.)

Message	Meaning
INCORRECT FID	The fid specified exceeds the field maximum for name (15 characters) or account (8 characters) or password (8 characters).
INSUFFICIENT MEMORY TO READ MAX RECORD OF FID	DRSP has failed to acquire enough memory to read the largest record of the load module specified as fid.
INSUFFICIENT MEMORY TO READ TREE	Memory space available to user is not sufficient to process the load module specified in the ENTER or REPLACE commands.
INSUFFICIENT OVERLAY SLOTS	The specified load module requires more processor overlay table entries than exist in the processor name table. This check on the table occurs during the first scan of the processor name table.
INSUFFICIENT PRIVILEGE FOR DRSP USAGE	The user must have a privilege level of 80 or greater to execute any DRSP commands.
INSUFFICIENT PRIVILEGE LEVEL TO PROCESS THIS COMMAND	The user does not have sufficient privilege of C0 to process ENTER, REPLACE and DELETE commands.
INSUFFICIENT SPACE ON SWAP RAD	The disk space allotted for new or replaced load modules is too small for the load module specified.
INSUFFICIENT VIRTUAL MEMORY TO EXECUTE DRSP	There are not enough virtual pages to allow DRSP to access the monitor.
MONITOR OVERLAY CANNOT HAVE OVERLAYS	A monitor overlay must not have an overlay.
NO ERRORS	No errors were encountered during command execution.
NO PRONAME SLOTS AVAILABLE	The number of extra processor name table entries is exhausted.
NO SUCH OVERLAY/ PROCESSOR	The proname entered cannot be found in the overlay or processor tables.
ONLY ONE LEVEL OF OVERLAYS FOR SHARED PROCESSORS	When analyzing the load module TREE record, more than one level of processor overlay was indicated.
ONLY PROCEDURE IS ALLOWED IN A PROCESSOR OVERLAY	DRSP checks a load module specified as a monitor overlay for procedure only.
OVLY DATA EXCEEDS RANGE 8000-8BFF	A monitor overlay, defined as 'data only', must reside in memory locations 8000_{16} to $8BFF_{16}$.
OVLY LINK EXCEEDS TABLE LIMIT	A system error to be reported.
PROCESSOR OVERLAY SLOTS EXHAUSTED	There are not enough empty processor overlay locations in the name table to fill the load module requirement. This check on the name table occurs during the write to the swap disk.
PROCESSOR/OVERLAY ALREADY EXISTS	User tried to ENTER a processor or overlay name that exists in the table.

Table 54. DRSP Error Messages (cont.)

Message	Meaning
PRONAME IS ILLEGAL	Some routines cannot be entered or replaced with DRSP (e.g., XDELTA, RECOVER, GHOST1, ALLOCAT, M:DUMLM).
PRONAME REQUIRED	A program must be specified with the ENTER, REPLACE and DELETE commands.
RAD OVERFLOW	Disk space allotted for the shared processors is exhausted.
READ ERROR READING FID (COPY)	I/O error detected while trying to read the processor for the copy into the system account.
SWAP I/O ERROR (QUEUE)	I/O error detected while writing processor to the swap disk.
WRITE ERROR WRITING FID (COPY)	I/O error detected while trying to write the processor into the system account. The processor is entered/replaced on non-"PERM" basis.
WRITE RAD FILE I/O ERRORS	I/O error detected while writing the processor to the swap disk.

Table 55. DRSP Information Messages

Message	Meaning
DRSP HERE	Routine title typed when user first enters DRSP.
DRSP INHIBIT SET	Another user is manipulating the shared processor tables and prevents any other user executing the ENTER, REPLACE and DELETE commands. However, the LIST and LISTALL commands can be executed at any time.
fid NEEDS xxxx GRANULES	If DRSP cannot find sufficient disk space in any available slot, it feeds back to the user the number of granules required to enter/replace the new load module.
proname REPLACED IN RAD SLOT #x	While exercising the "PERM" option, the proname in slot #x has been replaced by the monitor overlay proname specified in the current command.
PRONAME FOUND ON RAD	The proname already exists in the disk version of the processor tables when DRSP tries to execute the ENTER, PERM option. The "PERM" function is completed for the new copy.
PRONAME NOT FOUND ON RAD	The proname cannot be found in the disk version of the processor tables when DRSP tries to execute the REPLACE, PERM option. The "PERM" function is completed for the new copy.
USERS ASSOCIATED	DRSP attempts to replace TEL or CCI but finds there are users associated. The message is repeated periodically as long as users remain associated.
WAIT OPTION IGNORED FOR MONITOR OVERLAYS	"W" option is not appropriate for monitor overlays.

Table 56. DRSP Command Summary

Command	Description
D[ELETE] proname ,[PERM]	Prevents further user association with a processor.
END	Exits normally from DRSP.
E[NTER] proname $\left[\begin{Bmatrix}\text{FROM}\\\text{WITH}\end{Bmatrix}\text{fid}\right]$ [, option][, option][, option]	Enters a new shared processor or monitor overlay into the system.
L[IST] $\begin{Bmatrix}\text{proname}\\\#\text{xx}[-\text{yy}]\end{Bmatrix}$	Lists the processor name, the name table index, and the amount of disk space occupied by the processor.
LISTALL $\begin{Bmatrix}\text{proname}\\\#\text{xx}[-\text{yy}]\end{Bmatrix}$	Lists each shared processor name and its entries in certain tables.
R[EPLACE] proname $\left[\begin{Bmatrix}\text{FROM}\\\text{WITH}\end{Bmatrix}\text{fid}\right]$ [, option][, option][, option]	Replaces an existing shared processor or monitor overlay with a new shared processor or monitor overlay.
?	Requests a detailed error message when an error has been noted by DRSP.

11. ERROR MESSAGE FILE

INTRODUCTION

The error messages for the CP-V monitor and several CP-V processors are contained in an error message file, called ERRMSG. This file is initially created either through punched card or on-line terminal input and is maintained through use of the Edit processor. This chapter describes the structure of the ERRMSG file and the techniques required to create and modify the file.

Codes for detected error conditions are recorded in the job information table (JIT). The error code is placed in J:ABC (high-order byte) and the subcode is placed in ERO (right-justified). When CCI (batch jobs) or TEL (on-line jobs) is entered, a message is printed to correspond to the code and subcode. This message is obtained from the error message file (ERRMSG) via a keyed read using a key constructed from the group code, error code, and subcode. If either the file or the record corresponding to the code is missing, the error code itself will be printed. Otherwise, the message and the error code will be printed.

FORMAT OF ERROR MESSAGE FILE

Each record in the error message file contains the EBCDIC text of one error message. The key of each record is one word long and has the form

03	GC	EC	SC

The first byte always contains 03, which is the count of bytes in the key. The second byte is the group code, the third is the error code, and the fourth is the error subcode.

Group codes presently assigned are

0	Monitor	5	CCI
1	PCL	6	DRSP
2	Loader	7	Batch
3	TEL	8	Analyze
4	Runner		

Messages in the file with group codes other than zero are not handled by the monitor itself. Error codes currently assigned within the monitor group are

0 – 7F	I/O error and abnormal codes
80 – 9F	COBOL error codes
A0 – BF	Other Monitor codes
C0 – FF	Unused

The meaning of the assigned codes are defined in CP-V/TS Reference Manual, 90 09 07, CP-V/BP Reference Manual, 90 17 64, and in the ANS COBOL/LN Reference Manual, 90 15 00.

CREATING ERROR MESSAGE FILE

The ERRMSG file is initially entered into the system either through a card reader or an on-line terminal at the central site. The procedures for each type of input are described below.

CARD READER INPUT

Card input of the error message file is handled by the Error Message File Writer (ERRMWR). This program reads cards, interprets the first six columns as a hexadecimal number, converts this number into a three-byte key, and writes the card image exclusive of trailing blanks as a keyed record in the ERRMSG file in the account under which ERRMWR is executed. This account should be :SYS for the system error message file.

The card format is

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

Hex. code			Text of Message
G C	E C	S C	

Example:

Assume that the message ILLEGAL OPCODE is to be placed in the error message file for the monitor error code AE. The group code and subcode in this case are both zero. Thus, the card for this message would be punched as follows:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Hex. code			Text of Message
0 0	A E	0 0	I L L E G A L O P C O D E

Keys generated by the ERRMWR program have the form

03	GC (col. 1-2)	EC (col. 3-4)	SC (col. 5-6)

During conversion of the key, leading blanks are treated as zeros. Nonhexadecimal letters result in output of a warning message and cause the card to be ignored. The card image is scanned from right to left to determine the rightmost non-blank character, and the count of characters is adjusted so that trailing blanks are not written. A new line character X'15' is appended to the message.

The message may be continued in column 1 of the following card by appending a continuation character (;) at the end of the message in the first card. Only two cards per message are allowed.

A card containing an asterisk in column 1 is a control card and is used to set the format of the record written in the file. If column 2 of the control card contains a 0, the message key is appended to the front of the message text and is included in the record. If column 2 of the control card contains a 1, the key is not included in the record text (this is the default condition). Control cards can be placed anywhere within the data deck except between continuation cards.

TERMINAL INPUT

Creating or modifying the error message file can be accomplished from the terminal by using Edit or ERRMWR.

Example 1: Using Edit

```
!BUILD MSG (RET)
    1.000 00AB00 THAT'S NO DEBUGGER!(RET)
    2.000 00AB01 THAT'S NO OP CODE(RET)
    3.000(RET)
!SET M:EI DC/MSG (RET)
!ERRMWR (RET)
```

Example 2: Using ERRMWR

```
!SET M:EI UC(RET)
!ERRMWR(RET)
≥00AC01 DON'T ISSUE CAL3 OR CAL4(RET)
≥(RET)
!
```

12. HARDWARE ERROR LOG DISPLAY

INTRODUCTION

All hardware malfunctions occurring during system operation, whether recovered or not, are recorded in a special disk storage file. This file is periodically copied to two standard files (ERRFILE and SUMFILE) by a ghost program (ERR:FIL) that is initiated automatically for that purpose.

The resulting files may be listed and summarized by the two programs (ERR:LIST and ERR:SUM) that are described in this chapter. These files are also available for on-line preventive maintenance of the system and for diagnosis and prediction of hardware malfunctions.

ERR:LIST PROGRAM

The ERR:LIST program examines the error file (ERRFILE) for malfunction records that were written during the specified time period and produces a formatted listing of these records with (optionally) a summary of the records for that period. The formatted listing is complete with headings and formatting necessary for easy reading and use by field personnel.

The ERR:LIST program may be used in either on-line or batch mode. In on-line mode, control communications (both input and output) are directed to the terminal. In batch mode, control communications are input through the card reader and output through the line printer.

ERR:LIST can be run only under an account with A0 privilege.

There are three ERR:LIST commands: LIST, END, and HELP. The function and format of each of these commands is described below.

LIST This command lists the error records (and/or a summary of those records) that were entered into the ERRFILE after a specified time and/or date. The format of the command is

```
     {ERRORS    }  [{ON  } {ME   }]
LIST {SUMMARY   }  [{TO  } {M:LO }]
     {ERRORS AND}  [{OVER} {M:DO }]
     { SUMMARY  }  [{INTO} {M:EO }]
                   [       {fname}]

  [{AFTER    }                         ]
  [{BEGINNING} {mm/dd/yy hh:mm}        ]
  [{SINCE    } {hh:mm         }        ]  [AND END]
  [{STARTING } {mm/dd/yy      }        ]
  [{FROM     }                         ]
```

where

ERRORS specifies a listing of the error records in the ERRFILE.

SUMMARY specifies a summary of the number of errors by type and device.

ON, TO, OVER, INTO specify that the error records are to be output to the specified file or device.

ME specifies that output is to go to the on-line terminal through which the LIST command is entered.

M:LO specifies that output is to go to the device assigned to the LO DCB.

M:DO specifies that output is to go to the device assigned to the DO DCB.

M:EO specifies that output is to go to the device assigned to the EO DCB.

fname specifies the name of a file into which the output of the LIST command is to be written. The fname may specify either an old file or a new file and may be from 1 to 11 characters in length. Names larger than 11 characters are truncated at the right end.

AFTER, BEGINNING, SINCE, STARTING, FROM specify that error records entered into ERRFILE since the specified time are to be listed.

mm/dd/yy specifies the month (mm), day (dd), and year (yy) the listing is to start. All leading zeros must be included. The default date is the current data as returned by M:TIME. (Reference: CP-V/BP Reference Manual, 90 17 64.)

hh:mm specifies the hour (hh) and minutes (mm) the list is to start. Leading zeros must be included. The default time is 00:00.

AND END specifies that the program is to terminate after the listing is complete.

The default device is the terminal for on-line mode and the line printer for batch mode. Terms in a LIST command need not be entered in the order shown. ERR:LIST is insensitive to permutation of command phrases and extraneous words.

Examples:

>LIST ERRORS (RET)

(This causes all error records of the current date to be listed at the terminal.)

>LIST ERRORS AND SUMMARY

(This causes all error records of the current date to be listed at the terminal followed by a summary of errors by type and device.)

Note: If an output file is specified either directly or indirectly as a DCB name, it is written with keys acceptable to Edit. That is, the first line starts with key 1.000 and the increment is 1.000.

If ERRFILE cannot be opened or no records can be found in the date and time range specified, the following message will be typed:

```
THERE ARE NO RECORDS FOR THIS PERIOD
```

If ERRFILE does not exist, the following message will be typed:

```
ERR:FILE, :SYS DOES NOT EXIST
```

Other error messages may be typed during the processing of ERRFILE. See Appendix B of the CP-V/TS Reference Manual, 90 09 07.

END This command terminates the ERR:LIST program and returns control to the executive level (CCI or TEL). The format of the command is

END

HELP This command prints a sample of the correct command format for LIST. The format of the command is

HELP

ERR:SUM PROGRAM

ERR:SUM can be used only by an on-line user with a privilege level of A0. Should a user without this privilege level try to use ERR:SUM, the following message will be output to the terminal:

```
ERR:SUM ABORT:INSUFFICIENT PRIVILEGE
```

ERR:SUM is called by entering the name of the program in response to a prompt for commands by TEL.

Example:

!ERR:SUM

(Error summary.)

When initiated, ERR:SUM attempts to open the master error summary file (SUMFILE) which is produced by the automatic error file copying program. If the file does not exist, ERR:SUM so informs the user.

```
SUMMARY FILE DOES NOT EXIST
```

It then exits to TEL.

If the file cannot be opened for some other reason, ERR:SUM sends the following message to the terminal:

```
SUMMARY FILE CANNOT BE OPENED
```

It then returns to TEL.

After SUMFILE has been opened, ERR:SUM attempts to read the file. If an error occurs, ERR:SUM informs the user by typing

```
SUMMARY FILE READ ERROR
```

It then exits to TEL.

If no read error occurs, the error counts of the last summary record are added to the error counts of the master summary record and the summary (Figure 78) is printed on the LO device. This device may be the terminal or a line printer, depending on the assignment.

```
                MASTER ERROR LOG SUMMARY HH:MM MON DD, 'YY THROUGH HH:MM MON DD, 'YY

DEVICE       SIO FAILURE    DEVICE TIMEOUT    DEVICE ERROR    DEVICE FAILURE    TOTAL

yyndd            xx              xx               xx               xx            xxx
yyndd            xx              xx               xx               xx            xxx
  .              .               .                .                .              .
  .              .               .                .                .              .
yyndd            xx              xx               xx               xx            xxx
ERRLOG FILE COPYING ERRORS                                                       xxx
UNEXPECTED INTERRUPTS                                                            xxx
NO INTERRUPT RECOGNITION                                                         xxx
MEMORY PARITY ERROR                                                              xxx
SYSTEM STARTUP/RECOVERY                                                          xxx
WATCHDOG TIMER TRAP                                                              xxx
SOFTWARE DETECTED FILE INCONSISTENCY                                             xxx
SOFTWARE DETECTED SYMBIONT INCONSISTENCY                                         xxx

TOTAL            xxx             xxx              xxx              xxx           xxxx
```

Figure 78. Sample Error Summary Printout

After the master summary has been printed, ERR:SUM asks the user if SUMFILE is to be deleted.

```
PURGE SUMMARY FILE?
```

If the response is NO, SUMFILE is closed with a SAVE option and the following message is typed:

```
SUMMARY FILE SAVED.
```

ERR:SUM then exits to TEL.

If the user responds YES, confirmation is requested.

```
ARE YOU SURE?
```

If the user responds YES to this message, SUMFILE is closed with a REL option and the following message is typed:

```
SUMMARY FILE PURGED.
```

The summary file will be released only if ERR:SUM is running under the :SYS account.

ERR:SUM then exits to TEL.

If the user responds NO to "ARE YOU SURE?", he is again asked

```
PURGE SUMMARY FILE?
```

and the series of messages described above are repeated.

ERROR MESSAGES

Error messages for ERR:LIST and ERR:SUM are outlined in Table 57.

ERR:LIST COMMAND SUMMARY

Table 58 is a summary of ERR:LIST commands.

Table 57. ERR:LIST and ERR:SUM Messages

Message	Meaning
ARE YOU SURE?	ERR:SUM requests confirmation that SUMFILE should be purged.
ERR:FILE, :SYS DOES NOT EXIST	ERRFILE does not exist.
ERR:SUM ABORT:INSUFFICIENT PRIVILEGE	A user without A0 privilege attempted to use ERR:SUM.
PURGE SUMMARY FILE?	ERR:SUM questions whether SUMFILE should be purged.
SUMMARY FILE CANNOT BE OPENED	SUMFILE cannot be opened.
SUMMARY FILE DOES NOT EXIT	ERR:SUM cannot find SUMFILE.
SUMMARY FILE PURGED	SUMFILE has been purged following a YES response to a request for confirmation.
SUMMARY FILE READ ERROR	A read error occurred when SUMFILE was read.
SUMMARY FILE SAVED	SUMFILE has been saved following a NO response to a request for confirmation.
THERE ARE NO RECORDS FOR THIS PERIOD	ERRFILE cannot be opened by ERR:LIST or there are no error records for the time period requested.

Table 58. ERR:LIST Command Summary

Command	Description
END	Terminates the ERR:LIST program and returns control to the executive level.
HELP	Prints a sample of the correct format for the LIST command.
LIST {ERRORS | SUMMARY | ERRORS AND SUMMARY} [{ON | TO | OVER | INTO} {ME | M:LO | M:DO | M:EO | fname}] [{AFTER | BEGINNING | SINCE | STARTING | FROM} {mm/dd/yy hh:mm | hh:mm | mm/dd/yy}] [AND END]	Lists the error records in the ERRFILE and/or a summary of records after a specified starting time and/or date. ERRORS specifies a listing of errors. SUMMARY specifies a summary by type and device. ME specifies the on-line terminal. M:LO specifies output to the device assigned to the LO DCB. M:DO specifies output to the device assigned to the DO DCB. M:EO specifies output to the device assigned to the EO DCB. fname specifies the name of a file into which the errors and/or summary are to be written. mm/dd/yy specifies month, day, and year the list is to start. hh:mm specifies the time the list is to start. AND END specifies the program is to terminate after the listing is complete.

ERR:FIL PROGRAM

ERR:FIL copies the special file created by ERRLOG onto a normal keyed file (ERRFILE) that is more readily available to diagnostic programs. While copying ERRLOG's file, ERR:FIL compiles a summary of the errors copied. A file of these summaries (SUMFILE), which contains a summary of errors for each hour of operation and a master summary of all errors, is also maintained by ERR:FIL.

ERR:FIL is a ghost job that is awakened by ERRLOG whenever five errors have been recorded. ERR:FIL may also be awakened by a program with diagnostic privilege by using the initiate job CAL (CAL1,6 FPT) or by an operator key-in of GJOB ERR:FIL. ERR:FIL may also be run by an on-line user running under account :SYS with diagnostic privilege by typing !ERR:FIL.

The special file written by ERRLOG and the core buffers of ERRLOG, which are also input to ERR:FIL, are described in the following sections.

ERRFILE FORMATS

ERRFILE is a keyed file built and updated by ERR:FIL for use by diagnostic programs. The file contains one record for each error entry in the file created by ERRLOG.

The format of each record is identical to the format of the error entries in ERRLOG's file (ERRFILE). The keys for this file contain the Julian date in packed decimal, the time of the error in EBCDIC, and a sequence number for errors with the same time tag. This sequence number is reset to zero for each entry with a new time tag. The format of the key is

08	yy	0d	dd
h	h	m	m
n			

where

08 is the number of bytes in the key.

yy0ddd is the Julian date in packed decimal.

hhmm is the time (hours and minutes) in EBCDIC.

n is the sequence number.

The first record of ERRFILE is the key of the last record in ERRFILE and has a key of zero.

While copying records into ERRFILE, consistency and error checks are made on the input data. If any errors or inconsistencies are found, "copy error" records are written and a "copy error" counter in the summary record is incremented. The error and consistency checks, recovery actions taken, and the format of the copy error records are described below. The terminology used in the error record formats is defined in Table 59.

Table 59. Error Record Terminology

Term	Meaning
Account	The doubleword used to identify a user's collection of files.
AIO CC	A 4-bit field representing the condition codes as returned by the hardware in response to an AIO instruction.
AIO Status	A 16-bit field representing the status as returned by the hardware in response to an AIO instruction.
Alternate I/O Address	A 16-bit value representing an alternate physical I/O address by which a dual-access device can be referenced.
Bytes Remaining	A 16-bit field representing the Remaining Byte Count (RBC) field as returned by the hardware in response to a TDV instruction.
Consecutive, Keyed, Random	Methods of organizing user files in CP-V (refer to the CP-V/BP Reference Manual, 90 17 64).
Count of Entries Identical to Previous Entry	The number of error log records which are identical to one previously logged for identical reasons (excludes time records).
Count of Entries Lost	The number of error log records lost when logging becomes temporarily impossible for any reason.
Current Command Doubleword	A 64-bit value representing the command doubleword currently being processed for a device (indicated by the TDV status DW).
Caller's Address	The address back to which the error logging routine is returned when logging is complete; used in isolating software faults.
DCT Index	The 8-bit value indicating the order in which the device is configured into the system (at SYSGEN).
DCT Index of Symbiont Device	The 8-bit value indicating the order in which the device associated with the symbiont is configured into the system (at SYSGEN).
Effective Address	A 32-bit value representing the final address computed for the instruction pointed to by the instruction address (IA) in the PSD.
Error Subcode	An 8-bit field indicating which of several types of file inconsistencies has occurred (see CP-V/BP Reference Manual, 90 17 64).

Table 59. Error Record Terminology (cont.)

Term	Meaning
File Name	The TEXTC name used to identify a collection of user data on secondary storage.
Granule	The unit of secondary storage allocation equal to 2048 bytes (usually 2 sectors).
HIO CC	A 4-bit value (bits 0 – 3 of designated byte) representing the condition codes as returned by the hardware in response to an HIO instruction.
HIO Status	A 16-bit value representing the status as returned by the hardware in response to an HIO instruction.
I/O Address	A 16-bit value representing the physical I/O address.
I/O Count	A 32-bit value representing twice the number of SIO instructions executed for the device.
Julian Day	A 16-bit value representing the Julian day of the year (e.g., March 1 would be represented as X'3D') when the error was logged.
Length	An 8-bit value in the second byte of the error log record representing the number of useful 32-bit words contained in the error log record. It includes the first word in the count.
Memory Status Words (Sigma 9 only)	Each word is a 32-bit value representing data returned by the hardware in response to an LMS instruction.
MFI (Sigma 6 or 7 only)	A 4-bit value representing the current state of the memory fault indicators returned by the hardware in response to an RD instruction. All memory fault indicators will be reset.
Mode	A 16-bit value representing the manner in which the file was last referenced (see CP-V/BP Reference Manual, 90 17 64).
Model Number	A 16-bit value representing the conversion of a number (assigned by Field Engineering to uniquely identify peripheral devices) to a binary value (e.g., 7242 would be represented as X'1CA').
Number of Parity Errors	A 16-bit value representing the number of bad locations causing memory parity errors (only the first 14 bad locations are entered in the log if the number of errors is greater than 14).
Primary I/O Address	A 16-bit value representing the physical I/O address by which a device can be referenced (see Alternate I/O Address).
PSD	A 64-bit value representing the program status doubleword.
Real Address	A 32-bit value representing the actual memory address (in a mapped system, this is the same as the address in the IA field of the PSD).
Recovery Count	An 8-bit value initialized to zero at system initialization and incremented by the value one for every system recovery.
Relative Sector Address	A sector is 256 words. Each sector on a given device is numbered zero through device end. CP-V maintains file pointers by relative sector number, thereby simplifying the logic necessary to address different devices.
Relative Time	A 32-bit value representing milliseconds since midnight. Resolution is 2 msec.

Table 59. Error Record Terminology (cont.)

Term	Meaning
Relative Time Resolution	An 8-bit value, n, such that actual relative time resolution = 2 msec. (e.g., n = 1 for a resolution of 500HZ or 2 msec.).
Retries Remaining	An 8-bit value representing Retry Request minus the number of entries attempted. The range is between Retry Request and -1. A value of -1 indicates the operation was terminated due to retry count rundown.
Retry Request	An 8-bit value representing the maximum number of retries after which a device error is returned to the requester. This value is obtained from the requester's DCB.
Screech Code	The code used by CP-V to identify the system failure which has occurred.
Screech Subcode	An 8-bit field identifying which type of a specific and similar set of system failures has occurred. (See Software check codes in the CP-V/OPS Reference Manual, 90 16 75.)
Seek Address	The physical disk address last used to access this device.
Sense Information	The diagnostic information returned from the device as a result of sending a "sense" order to the device.
SIO CC	A 4-bit value (bits 0 - 3 of designated byte) representing the condition codes as returned by the hardware in response to an SIO instruction.
SIO Status	A 16-bit value representing the status as returned by the hardware in response to an SIO instruction.
Site Identification	A 64-bit field; the first 32 bits contain the EBCDIC representation of the SYSGEN input parameter for version; the second 32 bits contain blanks (X'40').
Startup Type	An 8-bit field indicating which of several types of system initialization was used.
Subchannel Status	The status of the I/O subchannel received from the hardware as a result of a TDV instruction.
Symbiont File	A CP-V system special file for buffering data between the CPU and slower speed line printers, card punchers, etc.
TDV CC	A 4-bit value (bits 0 - 3 of designated byte) representing the condition codes as returned by the hardware in response to a TIO instruction.
TDV Current Command DA	A 24-bit field representing the current command doubleword address used in obtaining the device status with a TDV instruction.
TDV Status Doubleword	A 24-bit field representing the subchannel status, as current command doubleword, device status, and byte count as returned by the hardware in response to a TDV instruction.
TIO CC	A 4-bit value (bits 0 - 3 of designated byte) representing the condition codes as returned by the hardware in response to a TIO instruction.
TIO Status	A 16-bit value representing the status as returned by the hardware in response to a TIO instruction.
Trap CC	A 4-bit value (bits 0 - 3 of designated byte) representing the condition codes as returned by the hardware when certain traps occur.

Table 59. Error Record Terminology (cont.)

Term	Meaning
Trapped Instruction	A 32-bit value representing the contents of the location pointed to by the instruction address (IA) in the PSD.
Type	An 8-bit value in the first byte of the error record which identifies the type of record.
Unit Address	A 6-bit value (bits 2 - 7 of designated byte) representing the address by which a processor can be referenced; the value is composed of a 3-bit cluster number followed by a 3-bit unit number.
Unit Type	An 8-bit value specifying the type of processor. Bit 0 of the designated byte indicates the presence of the processor in the current operational configuration (0 = present, 1 = not present).
User ID	A 16-bit value which is a unique number assigned by the system to the particular job or on-line session.
User Number	An 8-bit value which is the index into internal system tables used to access user specific information.
Version	The version identifier of the system running (i.e., A00, B00, etc.). This field is one byte in length. The letter of the version is stored in the first four bits and the number of the version is stored in the second four bits.
Volume Serial Number	A 4- or 6-byte field supplied by a user to identify either a tape or private pack.
Year	A 16-bit binary value representing the current year minus 1900 (e.g., 1973 is represented as X'49').

READ ERROR

If the condition codes set by T:RDERLOG indicate a read error, a copy error record (Read Error) is written and copying of the record is attempted. If inconsistencies are found in the record, a copy of the bad record is placed in the ERRFILE file, followed by the End Read Error record. If no inconsistencies are found, the record is processed normally and the Read Error record remains in the ERRFILE file. The record formats are

Read Error

word 0

Type X'10'	Length X'02'		Subtype X'01'
0 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

word 1

Relative time
0 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

End Read Error

word 0

Type X'10'	Length X'02'		Subtype X'02'
0 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

word 1

Relative time
0 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

ERRLOG RECORD LENGTH ERROR

If the length of the ERRLOG record is greater than 256, a copy error record followed by the ERRLOG record is written on ERRFILE. No attempt is made to copy this record in the detailed format. The record format is

word 0

Type X'10'	Length X'03'		Subtype X'03'
0 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

word 1

Relative time
bits 0–31

word 2

Index to bad entry
bits 0–31

INCORRECT TIME

If the time of an entry is out of sequence, i.e., if it is earlier than the time of the last record and the data has not changed, a copy error record is written on ERRFILE followed by the ERRLOG record. The time of this entry is then used for the key and processing continues. The record format is

word 0

Type X'10'	Length X'03'		Subtype X'05'
bits 0–7	bits 8–15	bits 16–23	bits 24–31

word 1

Relative time
bits 0–31

word 2

Index to bad entry
bits 0–31

where index is the displacement within the ERRLOG record of the first word of erroneous entry.

ILLEGAL ENTRY TYPE

If the entry type is not one of the legal types, a copy error record followed by the ERRLOG record is written on ERRFILE. No attempt is made to copy the remainder of the record. The record format is

word 0

Type X'10'	Length X'03'		Subtype X'06'
bits 0–7	bits 8–15	bits 16–23	bits 24–31

word 1

Relative time
bits 0–31

word 2

Index to bad entry
bits 0–31

where index is the displacement within the ERRLOG record of the first word of erroneous entry.

Note: Errors that occur while booting have a time tag of 24XX but the keys of these records contain the current date and 0011 for the time.

If read or write errors are detected while reading or writing ERRFILE and SUMFILE, they are ignored.

Whenever I/O errors or certain unusual conditions occur, an entry will be made into the ERRLOG file. This entry will contain any information pertinent to the condition.

Word 0 of each entry will have a code indicating which error or unusual condition is present along with the number of words in the entry (including word 0). Time (hhmm) and Device Name (yyndd) are in EBCDIC.

There are no error log entries for the following two interrupts.

MEMORY FAULT INTERRUPT

The Memory Fault Interrupt (MFI) is triggered when an error is detected during a memory access by either the CPU or an IOP. If the MFI is triggered by the CPU, a parity error trap will also occur unless the error is a Loop Check Parity error or Overtemperature condition. The parity error trap routine performs error recovery, logs the error, and clears the MFI to avoid duplicate processing. The MFI service routine therefore expects to only handle errors detected during an IOP memory access and Loop Check and Overtemperature errors. The Loop Check and Overtemperature errors are processed by the memory parity program and the system recovery program is entered with code X'23'. The other errors are logged by the device handler, which also performs the required recovery.

PROCESSOR FAULT INTERRUPT

The Processor Fault Interrupt is not enabled in CP-V. Errors that cause this interrupt in a monoprocessor system are handled by the I/O Interrupt Routines.

SIO FAILURE

This record is logged when CC1 and/or CC2 are set after execution of the SIO instruction.

word 0

Type X'11'	Length X'06'	Model number
bits 0–7	bits 8–15	bits 16–31

word 1

Relative time
bits 32–63

word 2

SIO status	I/O address
bits 0–15	bits 16–31

word 3

MFI (if Sigma 6 or 7)	SIO CC	TDV CC	
0–7	8–15	16–23	24–31

word 4

Subchannel status		TDV current command DA
0–7	8–11	12–31

word 5

TDV status	Bytes remaining
0–15	16–31

TIME OUT

This record is logged when the I/O interrupt does not occur within a specified time period in response to an I/O instruction. The format for time out (type X'12') is identical to that for device error (type X'15') except that HIO CC and status replaces AIO CC and status.

UNEXPECTED INTERRUPT

This record is logged when an interrupt, other than an attention interrupt, is received from a known device for which no I/O operations have been started by the system.

word 0

Type X'13'	Length X'04'	Model number
0–7	8–15	16–31

word 1

Relative time
0–31

word 2

AIO status	I/O address
0–15	16–31

word 3

	AIO CC	
0–7	8–15	16–31

DEVICE ERROR

This record is logged when general analysis of the status received from an AIO, TDV, or TIO indicates an error which resulted from the I/O operation.

word 0

Type X'15'	Length X'0D'	Model number
0–7	8–15	16–31

word 1

Relative time
0–31

word 2

AIO status	I/O address
0–15	16–31

word 3

MFI (if Sigma 6 or 7)	AIO CC	TDV CC	TIO CC
0–7	8–15	16–23	24–31

word 4

Subchannel status		TDV current command DA
0–7	8–11	12–31

word 5

TDV status	Bytes remaining
0–15	16–31

words 6 and 7

Current command doubleword
0–31

word 8

TIO status	Retry request	Retries remaining
0–15	16–23	24–31

word 9

Total I/O count for this device
0–31

words 10 and 11

Volume serial number (Six characters for ANS tapes)	Original function code	Current function code
0–15	16–23	24–31

word 12

Seek address
0–31

The format for device error (type X'15') is identical to that for device timeout (type X'12') except that HIO CC and status is used in place of AIO CC and status.

SECONDARY RECORD FOR DISK PACK, RAD, TAPE, AND RBT DEVICE ERRORS

This record is generated as a result of a previous device error and contains device status which corresponds to the information contained in the Device Error record (type X'15') preceding this record.

word 0

Type X'16'	Length (variable)	I/O address

word 1

Relative time

words 2 and following

Sense information (left justified)

Device	Length (in bytes)
7242	10
Disk A, B	16
1600 bpi tape	6
9T tape	1
RAD	3
7670 RBT	1

The I/O address links the secondary record to the corresponding device error entry.

HARDWARE ERROR

This record is logged when a hardware error has been detected, the type of error being indicated by the Trap CC. For Sigma 6 and 7, this record is generated as a result of the memory parity interrupt associated with location X'56'. For Sigma 9, this record is generated as a result of the parity error trap associated with location X'4C'.

word 0

Type X'17'	Length X'09'		Trap CC

word 1

Relative time

words 2 and 3

PSD word 1
PSD word 2

words 4 and 5

Reserved for future use

word 6

Trapped instruction

word 7

Effective address of trapped instruction; 0 if indeterminate

word 8

Real address of trapped instruction

SYSTEM STARTUP

This error is logged at system initialization and at every recovery.

word 0

Type X'18'	Length X'04'	Startup type	Recovery count

word 1

Relative time

word 2

Year (last two digits; in binary)	Julian day (in binary)

word 3

Screech code	Screech subcode	

where

screech code and screech subcode are defined in the CP-V/OPS Reference Manual, 90 16 75.

recovery count is set to 0 for initial startup as defined by startup types 1, 2, or 3 below.

startup type   specifies the type of startup.

1 – Initial PO boot

2 – PO boot with files

3 – System device boot (no recovery)

4 – System recovery

5 – Operator recovery

WATCHDOG TIMER

This record is generated as a result of the instruction watchdog timer runout trap associated with location X'46'.

word 0

Type X'19'	Length X'09'		Trap CC

word 1

Relative time

words 2 and 3

PSD word 1
PSD word 2

words 4 and 5

Reserved for future use

word 6

Trapped instruction

word 7

Effective address of trapped instruction; 0 if indeterminate

word 8

Real address of trapped instruction

FILE INCONSISTENCY ERROR

This record is logged if the system detects files which are inconsistent in that the associated file links do not match or are otherwise incorrect.

word 0

Type X'1A'	Length X'14'	75	Error subcode

word 1

Relative time

words 2 through 9

File name

words 10 and 11

Account number

word 12

DCT index	Relative sector address

word 13

ORG	MODE

where

ORG   is set to 1 for consecutive, 2 for keyed, and 3 for random.

MODE   is set to 1 for IN, 2 for OUT, 4 for INOUT, and 8 for OUTIN.

CORRECTABLE SEEK ERROR

The correctable seek error entry is identical to the File Inconsistency record above except that it has an error subcode of 7F. This record is logged when a hardware seek error occurs and a subsequent retry clears the error.

SOFTWARE-DETECTED SYMBIONT INCONSISTENCIES

This record is logged if the system detects files which are symbiont inconsistent in that the associated file pointers do not match or are otherwise incorrect.

word 0

Type X'1B'	Length X'03'	0	DCT index of symbiont
0–7	8–15	16–23	24–31

word 1

Relative time
0–31

word 2

Code	DCT index	Relative sector number
0–7	8–15	16–31

where the code field has the following meanings:

X'F1' – bad disk address.

X'F2' – bad backward link.

X'F3' – file device error.

X'F4' – bad record control code.

X'F5' – data overruns block.

X'F6' – record device error.

X'F7' – file device down.

SIGMA 9 INSTRUCTION EXCEPTION

This record is logged as a result of the instruction exception trap associated with location X'4D'.

word 0

Type X'1D'	Length X'05'	0	Trap CC
0–7	8–15	16–23	24–31

word 1

Relative time
0–31

words 2 and 3

PSD word 1
PSD word 2
0–31

word 4

Trapped instruction
0–31

LOST ENTRY INDICATOR

This record is entered when buffering constraints make error logging temporarily impossible. The newest entries are lost.

word 0

Type X'1E'	Length X'02'	Count of entries lost
0–7	8–15	16–31

word 1

Relative time of last lost entry
0–31

DUPLICATE ENTRIES

This record is logged if duplicate error log entries are generated.

word 0

Type X'1F'	Length X'02'	Number of entries identical to previous
0–7	8–15	16–31

word 1

Relative time of last duplicate
0–31

POWER ON

This record is generated as a result of the power on trap associated with location X'50'.

word 0

Type X'20'	Length X'02'	
0–7	8–15	16–31

word 1

Relative time of power off
0–31

CONFIGURATION RECORD

This record is logged at system startup. The time field in the entry key contains 0010.

word 0

Type X'21'	Length (variable)	
0–7	8–15	16–31

word 1

Relative time
0–31

word 2

Model number		DCT index

word 3

Alternate I/O address	Primary I/O address

additional words

Words 2 and 3 may be repeated up to four times. The pairs of words will be in order by DCT index. Multiple records may occur.

SYSTEM IDENTIFICATION

This record is entered at system startup and recovery and is entered after the CONFIGURATION RECORD (type X'21'). The time field in the entry key also contains 0010.

word 0

Type X'22'	Length X'05'	Core size (in 8K word blocks)	Relative time resolution

word 1

Relative time

word 2

System, version, flags (loc 2B)

words 3 and 4

Site identification (in EBCDIC)

TIME STAMP

This record is entered one each hour on the hour.

word 0

Type X'23'	Length X'03'	

word 1

Relative time

word 2

Year (last two digits; in binary)	Julian day (in binary)

BAD GRANULE RELEASE

This record is logged if the granule being released contains an invalid disk address or has already been released (dual allocation).

word 0

Type X'24'	Length X'04'	Type code (0–bad address; 1–dual allocation)

word 1

Relative time

word 2

Relative sector address

word 3 (if type code = 0)

Address of routine calling ERRLOG

or

word 3 (if type code = 1)

Number of graules being released	

OPERATOR MESSAGE

This record is interjected as the result of an operator ERRSEND key-in or by a diagnostic program. It is generally used to describe unusual conditions surrounding a particular error.

word 0

Type X'27'	Length (variable)	

word 1

Relative time

word 2

Message from the operator in TEXTC format. (Maximum size is 71 characters plus the count byte.)

I/O ACTIVITY COUNT

This is recorded once per hour and at recovery.

word 0

0–7	8–15	16–31
Type X'28'	Length (variable)	DCT index of first device

word 1

0–31
Relative time

word 2

0–15	16–31
I/O address	DCT index

word 3

0–31
I/O count

additional words

0–31
Words 2 and 3 may be repeated up to four times. The pairs of words will be in order by DCT index. Multiple records may occur.

MFI PRIMARY RECORD

This record is logged as a result of the memory fault interrupt associated with location X'57' on a Sigma 9.

word 0

0–7	8–15	16–31
Type = X'31'	Length X'02'	

word 1

0–31
Relative time

MEMORY PARITY SECONDARY RECORD

This record is logged as a result of the memory fault interrupt associated with location X'57' or the memory parity trap associated with location X'4C' on the Sigma 9. This record follows record type X'17' and record type X'31'.

word 0

0–7	8–15	16–31
Type X'43'	Length X'05'	

word 1

0–31
Relative time

word 2

0–31
Memory status word 0

word 3

0–31
Memory status word 1

word 4

0–31
Memory status word 2

ENQUEUE TABLE OVERFLOW

This record is logged when an Enqueue CAL has been rejected because there are insufficient unused entries in the Enqueue tables (QT). The user with the most entries in the QT is specified.

word 0

0–7	8–15	16–31
Type X'50'	Length X'03'	

word 1

0–31
Relative time

word 2

0–15	16–31
User ID	Entry count (in binary)

Entry count is the number of entries in the enqueue table belonging to the specified user at the time the error log entry was made.

PARTITIONED RESOURCE

This entry is logged when a resource is partitioned via the SYSCON processor by the operator.

word 0

0–7	8–15	16–31
Type X'51'	Length X'03'	Model number

word 1

0–31
Relative time

word 2

0	1–15	16–31
F	0———————0	I/O address

F = 0 for device entry.
F = 1 for controller entry.

RETURNED RESOURCE

This entry is logged when a resource is returned from being partitioned via the SYSCON processor by the operator.

word 0

0–7	8–15	16–31
Type X'52'	Length X'03'	Model number

word 1

Relative time
bits 0–31

word 2

F	0————————0	I/O address
0	1–15	16–31

F = 0 for device entry.

F = 1 for controller entry.

PRIVATE PACK INCONSISTENCY

This record is logged as a result of a logic error in the control of access to the private pack other than file inconsistency (which is reported by a type X'1A' record). This error is not related to file pointer errors but results from a system error.

word 0

Type X'53'	Length X'04'	(shaded)
0–7	8–15	16–31

word 1

Relative time
bits 0–31

word 2

User ID	DCB address
0–7	8–31

word 3

Address of routine calling ERRLOG
bits 0–31

HARDWARE-ERROR DIAGNOSTIC CALs

The following four CALs are intended for use by the monitor in performing diagnostic functions relating to the hardware-error log and must be issued by a program from the :SYS account. They provide the following services: reading from the hardware-error log, writing to the hardware-error log, initiation of diagnostic ghost jobs, and accessing of symbiont devices directly for diagnostics.

These four services are all invoked by a CAL1,6 fpt instruction; the addressed FPT contains a code and a parameter. The FPT codes and the functions performed are as follows:

FPT Code	Function
0	Read Error Log
1	Write Error Log
6	Initiate Ghost Job

The status of the requested operation is reported via condition-code settings summarized below. (Not all of the statuses indicated are appropriate to, or reported by, all three CALs.)

CC1	CC2	CC3	CC4	Status
0	0	0	0	Normal return.
1	0	0	0	Request denied: insufficient privilege, not in :SYS account, or buffer is not a data page.
0	1	0	0	Error during operation (Read or Write), or job unknown (Initiate).
0	0	1	0	Last buffer.
0	0	0	1	Error log does not yet exist (Read).

In each case, the calling program must be of privilege level C0 or greater; otherwise CC1 is set to 1 and no action is taken.

READ ERROR LOG

The format of the FPT for a read-error-log request is

X'00'	(shaded)	Buffer address
0–7	8–15	16–31

A variable number of words up to a maximum of 256, depending upon the contents of the error log, is read to the area addressed by the FPT. This is a 'destructive' read, returning error-log granules to the monitor's available pool as they are exhausted.

The error-log file is not protected against simultaneous use; thus only one program in the entire system should read this file.

WRITE ERROR LOG

The format of the FPT for a write-error-log request is

X'01'	(shaded)	Buffer address
0–7	8–15	16–31

The second byte of the data record addressed by the FPT must specify the number of words to be written, up to a maximum of 253. The first byte of the record should contain a type code.

INITIATE GHOST JOB

The format of the three-word FPT for an initiate-job request is

word 0

0–7	8–31
X'06'	0————————————————0

words 1 and 2 (Name of job to be initiated)

0–7	8–15	16–23	24–31
n	a_1	a_2	a_3
a_{n-3}	a_{n-2}	a_{n-1}	a_n

(Name of job must be in TEXTC format.)

If the program to be initiated is already in execution at the time of the request and is not in a waiting state (WAIT CAL with unexpired time), the normal return is made (CCI=0). If the program is in a waiting state, it will be activated immediately at the WAIT CAL plus 1 and a normal return is made to the initiating program.

ACCESS DIRECTLY FOR DIAGNOSTICS

Diagnostics, peripheral exercisers, and other programs may gain direct connection to card readers, card punches, line printers, and magnetic tapes by using the following special form of the M:OPEN:

option device code (P14) alternate form

0–14	15	16	17–20	21–31
0——————0	0	1	0—0	I/O address

where I/O address is formatted as specified in the Sigma 7 Computer/Reference Manual, 90 09 50. This special form (diagnostic M:OPEN) can only be used to achieve connection to a symbiont device.

When devices are accessed in this way, the status given by TDV, TIO, and AIO instructions is returned in the DCB words 15-18 as shown below.

	0 — 15	16 — 31
15	0——————0	AIO status
16	0——————0	TDV command address
17	TDV status	TDV byte count
18	TIO status	TIO byte count

SUMFILE

SUMFILE is a keyed file built and updated by ERR:FIL for use by ERR:SUM. The first record of SUMFILE is the key of the last record of SUMFILE and has a key of zero. The second record of the file is the master error summary and has a key of one. Subsequent records of the file are the summaries of errors for each hour of operation. The keys of these records begin at two and are incremented for each successive record. The master error summary is the summation of the hour summaries. The format of the summary records (master and hourly) is detailed in Figure 79. The format of the keys for SUMFILE is

01	N

where

01 is the number of bytes in key.

N is the sequence number.

Word	Contents	Description
0	Length	Number of words in record.
1	h h : m	Time: The time of the first error included in this summary for the master summary and open summary records. For closed summary records, it is effectively the closing or completion time.
2	m b̸ m o	
3	n b̸ d d	
4	, ' y y	
5	ERROR	Count of "copy error" records.
6	UXI	Count of unexpected interrupt entries.
7	NIR	Count of unrecognized interrupt entries.
8	MPE	Count of memory parity error entries.
9	SYS	Count of system start-up entries.
10	WDT	Count of watchdog timer trap entries.
11	FIL	Count of file inconsistency entries.
12	SYM	Count of symbiont inconsistency entries.
13	PAR	Sigma 9 parity error – memory fault.
14	INX	Sigma 9 instruction exception error.
15	b b b y	EBCDIC device name.
16	y n d d	
17	SIO	Count of SIO failure entries for this device.
18	DTO	Count of Device Timeout entries for this device.
19	DE	Count of Device Error entries for this device.
20	DF	Count of Device Failure entries for this device.
21 ⋮ 26	⋮ Words 15-20 repeated for each device in DCT. ⋮	

Figure 79. Summary Record Format

13. SYSCON

INTRODUCTION

SYSCON is a system control processor that enables the following functions to be performed:

- A resource may be partitioned out of the CP-V system so that it is no longer available to the users (the only exception being a user who is using special peripheral diagnostic procedures). In general, this is used for removing faulty resources from the system. The following types of devices and their associated controllers are the only types of resources that can be partitioned.
  - Card reader
  - Card punch
  - Line printer
  - Magnetic tape drive
  - Private disk pack spindle
- A partitioned resource may be returned to the system.
- The status of the various system resources can be displayed.
- The M:MODNUM file can be built, updated, and displayed. The M:MODNUM file contains a list of all legitimate device/controller model number combinations for the CP-V installation.

SYSCON can be run either as a ghost job or as an on-line job. The user must have at least A0 privilege. SYSCON is called as a ghost job with the operator key-in

GJOB SYSCON

It is called as an on-line job with the command

SYSCON

If the user privilege level is sufficient, SYSCON displays the message

SYSCON HERE

and prompts for a command with the 'greater than' character (>).

Example:

!GJOB SYSCON (NL)

SYSCON HERE

>
.
.

SYSCON COMMANDS

DISPLAY The DISPLAY command displays the status of resources. The format of the command is:

```
            ⎡[yy]ndd              ⎤
            ⎢yy                   ⎥
DI[SPLAY]   ⎢CONT[,[yy]ndd]       ⎥
            ⎢PART                 ⎥
            ⎣ALL                  ⎦
```

where

[yy]ndd requests the status for the specified device. The yy portion is unnecessary and is therefore optional. However, if it is specified, it is verified by SYSCON. This request can only be made for partitionable types of devices (i.e., card readers, card punches, line printers, magnetic tape drives, and private disk pack spindles).

yy requests the status for all devices of type yy. This request can only be made for partitionable types of devices.

CONT[,[yy]ndd] requests the status of all controllers within the system if the [yy]ndd portion is omitted, or requests the status of just the controller for the device specified by [yy]ndd. In the latter case, the yy portion is unnecessary and is therefore optional. However, if it is specified, it is verified by SYSCON. When a device is specified, it must be a partitionable type of device.

PART requests a display of all partitioned resources (i.e., partitioned devices and controllers).

ALL requests a display of all resources within the system and their status. (The display includes every device and controller, not just those that are partitionable.)

If no parameter is specified, ALL is the default.

The general format of the display is as follows:

```
*************************************************
    RESOURCE            STATUS
-------------------------------------------------
                        ⎧PART                 ⎫
                        ⎪⎧IDLE ⎫              ⎪
    yyndd               ⎨⎨uid  ⎬  [NON-PART]  ⎬
                        ⎪⎩blank⎭              ⎪
                        ⎩                     ⎭

                        ⎧PART    ⎫
    CONT,yyndd          ⎨UP      ⎬
                        ⎩NON-PART⎭
*************************************************
```

where

yyndd specifies the device name.

CONT,yyndd specifies the controller associated with the device specified by yyndd.

PART indicates that the resource is partitioned.

IDLE indicates that the device is idle.

uid specifies a tape label or a disk pack serial number and indicates that the device is busy. (This is only applicable for a tape drive or a private disk pack spindle.)

blank indicates that the device is busy. This is only applicable for devices other than tape drives or private disk pack spindles.

UP specifies that the controller is available.

PARTITION The PARTITION command attempts to remove a resource from the system. It may not be possible to partition a resource because, for example, the resource may be nonpartitionable. If it is not possible to partition the requested resource, a message is output to indicate the reason why the resource was not partitioned. When a resource is successfully partitioned from the system, any user currently associated with it is allowed to access the resource until completion. However, no new user will be permitted association with the resource unless that user is using special peripheral diagnostic procedures. (In the case of a failing resource, the operator may want to abort the user(s) associated with the resource.) When a nonsymbiont type device is partitioned, the total count of devices for that type of resource is decremented by one and all jobs waiting in the queue are rescheduled according to the new count.

The format of the PARTITION command is:

PA[RTITION] {[yy]ndd | CONT,[yy]ndd}

where

[yy]ndd specifies the address of the device that is to be partitioned. The yy portion is unnecessary and therefore is optional. However, if it is specified, it is verified by SYSCON.

CONT,[yy]ndd specifies the name of a controller that is to be partitioned. When a controller is partitioned, all of its devices are also partitioned. If any of its devices cannot be partitioned for any reason, then the controller (and its devices) will not be partitioned. The yy portion of the specification is unnecessary and therefore is optional. However, if it is specified, it is verified by SYSCON.

RETURN The RETURN command attempts to return a partitioned resource to the system. It may not be possible to return a resource to the system because, for example, the resource may be busy. If the requested resource cannot be returned, a message is output to indicate the reason why the resource was not returned. When a resource is successfully returned to the system, it becomes available to users in the same manner as it was available before it was partitioned. When a nonsymbiont type device is returned, the total count of devices for that type of resource is incremented by one and all jobs in the queue are rescheduled according to the new count.

The format of the RETURN command is:

RE[TURN] {yy]ndd | CONT,[yy]ndd}

where

[yy]ndd specifies the name of a device that is to be returned to the system. The yy portion is unnecessary and therefore is optional. However, if it is specified, it will be verified by SYSCON.

CONT,[yy]ndd specifies the name of a controller that is to be returned. When it is requested that a controller be returned to the system and one or more devices within the controller are busy, the controller will be returned and the devices that are not busy will be returned. The busy devices will <u>not</u> be returned and will therefore remain partitioned. A message will be issued for each device indicating whether or not it was returned and each busy device must later be returned individually. The yy portion of the specification is unnecessary and therefore is optional. However, if it is specified, it will be verified by SYSCON.

MODNUM The MODNUM command is used to generate, update, or display the contents of the file M:MODNUM. M:MODNUM is a file that contains a list of all legitimate device/controller model number combinations within the CP-V system. This file is used during PASS2 of SYSGEN when processing the :DEVICE command. The device/controller model numbers are verified by SYSGEN and, if needed, the appropriate controller model number is obtained from the file.

Generally, the M:MODNUM file resides in the :SYS account. However, it may reside in another account provided that appropriate assignments are made. SYSCON expects the input M:MODNUM file to reside in the :SYS account, but this can be overridden by assigning the M:EI DCB to the account that the file actually resides in (before calling SYSCON). For example, assume that the file resides in the account MYACCT:

!SET M:EI/M:MODNUM.MYACCT (RET)

!SYSCON (RET)

Note that the file may reside in another account only when SYSCON is run in the on-line mode. Such an assignment cannot be made when SYSCON is run as a ghost.

When the M:MODNUM file is created or updated, the output M:MODNUM file will be output to the account in which SYSCON is run.

SYSGEN also expects to find the M:MODNUM file in the :SYS account; however this can be overridden by an appropriate assignment (see the :DEVICE command in the SYSGEN chapter).

The MODNUM command has the following format:

$$\text{MO[DNUM]}\begin{Bmatrix}\text{NEW}\\\text{UPDATE}\\\text{DISP}\end{Bmatrix}$$

where

NEW specifies that a new M:MODNUM file is to be generated. SYSCON will prompt for entries to the file as described below.

UPDATE specifies that the M:MODNUM file is to be updated (i.e., entries are to be added or deleted). SYSCON will prompt for updates as described below.

DISP specifies that the contents of the M:MODNUM file are to be displayed. An example of the format of the display is given in Figure 80.

```
*************************************************
  DEVICE                    CONTROLLER
-----------------------------------------------
   7120                       7120
    :                          :
   7242                       7240
    :                          :
*************************************************
```

Figure 80. Display of the M:MODNUM File

When the MODNUM command is given with the NEW or UPDATE option, SYSCON prompts for the input of subcommands with a period. The formats of the subcommands are:

I device model#, controller model#

$$\text{D}\begin{Bmatrix}\text{device model\#}\\\text{controller model\#}\end{Bmatrix}$$

STOP

where

I specifies that an entry for the device model number and its corresponding controller model number is to be inserted into the M:MODNUM file.

D specifies that all entries in M:MODNUM for the specified device model number or controller model number are to be deleted.

STOP terminates the set of MODNUM subcommands and causes the M:MODNUM data to be written to the M:MODNUM file.

device model# specifies the device model number. It must be a five character field. The first character must be a D followed by a four hexadecimal digit device model number.

controller model# specifies the controller model number and must consist of four hexadecimal digits.

The M:MODNUM file may have several entries in which unique device model numbers each have the same controller model number. Also, the file may have several entries where the same device model number has different controller model numbers associated with it.

At least one of the entries in the file must contain a default device/controller model number combination. This allows an installation with a nonstandard device to enter the device into the system during SYSGEN. This is useful when the model number is not known or is not acceptable (e.g., it has too many digits or contains digits that are not hexadecimal). The default entry is:

device model# = FFFF

controller model# = FFFF

END The END command terminates SYSCON. The format of the command is

END

SAMPLE SYSCON SESSION

Figure 81 is a sample on-line SYSCON session in which a line printer and a controller with five devices are partitioned.

The DISPLAY command is used to list all devices and controllers that are partitioned. All terminal output is underscored; terminal input is not.

```
!SYSCON (RET)
SYSCON HERE

≥PA LPAOF (RET)
SYMBIONT LPAOF TERMINATED
ITEM PARTITIONED

≥PA CONT,9TA80 (RET)
ITEM PARTITIONED
ITEM PARTITIONED
ITEM PARTITIONED
ITEM PARTITIONED
ITEM PARTITIONED

≥DI PART (RET)

**********************************
  RESOURCE              STATUS
 ------------------------------

1PAOF                PART
9TA80                PART
9TA81                PART
9TA82                PART
9TA83                PART
CONT,9TA80           PART
**********************************

≥END (RET)
```

Figure 81. Sample SYSCON Session

SYSCON PROCEDURES

M:DPART and M:DRET are two system procedures that were developed for use in the SYSCON processor. They are used to partition and return resources and require at least an A0 privilege level. However, they should never be used in any user-written program. Their description is included here only for completeness of documentation. Partitioning and returning resources requires some extremely important interface with the monitor. That is, appropriate checks must be made and certain restrictions must be strictly observed. Any program that uses these procedures may seriously affect the operation and integrity of the system.

M:DPART M:DPART is the system procedure used by SYSCON when a specific controller or device is to be removed from the system by partitioning. The M:DPART procedure call is of the form

M:DPART ({DEV / CONT}, [*]device address)

where

DEV specifies that a device is to be partitioned.

CONT specifies that a controller is to be partitioned.

device address has the form ndd in which n specifies the IOP unit address (the number associated with the IOP letter; see Table 60) and dd specifies the device number (see Table 61).

Table 60. IOP Designation Codes

Unit Address (n)	IOP Letter
0	A
1	B
2	C
3	D
4	E
5	F
6	G
7	H

Table 61. Device Designation Codes

Device Designation (dd)	Meaning
00 to 7F	Refers to a device number (00 through 7F).
80 to FF	Refers to a device controller number (8 through F) followed by a device number (0 through F).

Calls generated by the M:DPART procedure have the form

CAL1,6 fpt

where fpt points to word 0 of the FPT shown below.

word 0

*	X'0A'	Device address
0	1–7	15–31

word 1

0	0	f_1	0 ——————————————————— 0
0	1	2	3 – 31

where f_1 specifies either that a controller is to be partitioned (f_1=1) or that a device is to be partitioned (f_1=0).

The return from the procedure call is to CAL+1 with the following possible condition code settings:

CC1=0	partitioning completed unless CC2=1.
CC1=1	symbiont suspended and therefore not partitioned.
CC2=0	device address is valid; partitioning completed.
CC2=1	unknown device address; partitioning request ignored.

When a resource is partitioned, a record is entered in the error log file (ERRFILE). The record has type code X'51'.

M:DRET M:DRET is the system procedure used by SYSCON when a specific controller or device that is partitioned is to be returned to the system. The M:DRET procedure call is of the form

M:DRET ({DEV|CONT}, [*]device address)

where

DEV specifies that a device is to be returned.

CONT specifies that a controller is to be returned.

device address has the form ndd in which n specifies the IOP unit address (the number associated with the IOP letter; see Table 60) and dd specifies the device number (see Table 61).

Calls generated by the M:DRET procedure have the form

CAL1,6 fpt

where fpt point to word 0 of the fpt shown below.

word 0

*	X'0A'	Device address
0	1 – 7	8 – 31

word 1

0	1	f_1	0 ——————————————————— 0
0	1	2	3 – 31

where f_1 specifies either that a controller is to be returned (f_1=1) or that a device is to be returned (f_1=0).

The return from the procedure call is to CAL+1 with the following possible condition code settings:

CC1=0	return was successful.
CC1=1	return was unsuccessful because either the specified device was busy or one or more devices on the specified controller were busy. In the latter case, the controller and the nonbusy devices were returned.
CC2=0	device address is valid; return completed.
CC2=1	unknown device address; return request ignored.

When a resource is returned, a record is entered in the error log file (ERRFILE). The record has type code X'52'.

SYSCON MESSAGES

There are two types of messages that are output by the SYSCON processor. The first type (Table 62) is output on the terminal from which SYSCON is being run (either an on-line terminal or the operator's console). The second type (Table 63) is output on the operator's console to notify the operator that a resource has been partitioned or returned. The latter type of message only appears when SYSCON is not being run from the operator's console.

Table 62. SYSCON Messages for the User's Terminal

Message	Description
CANNOT MAP INTO MONITOR	SYSCON cannot be run because required tables of information cannot be made accessible to SYSCON. Contact the system manager.
*CANNOT OPEN M:MODNUM xxxx	An I/O error occurred when trying to open the M:MODNUM file for output. The error/abnormal code and subcode are specified by xxxx.

Table 62. SYSCON Messages for the User's Terminal (cont.)

Message	Description
*CANNOT OPEN M:MODNUM account xxxx	The M:MODNUM file does not exist in the specified account and therefore cannot be opened. The I/O error/abnormal code and subcode is specified by xxxx.
CANNOT PARTITION, ITEM ALREADY PARTITIONED	The specified resource is already partitioned. The command is ignored.
CANNOT PARTITION, ITEM NON-PARTITIONABLE	Either the resource specified is not partitionable or the resource is a symbiont that has been suspended. The command is ignored.
CANNOT PARTITION, ITEM NOT PRESENT	The resource specified does not exist in the system. The command is ignored.
*CANNOT READ M:MODNUM account xxxx	An I/O error occurred when trying to read the M:MODNUM file. The error/abnormal code and subcode are specified by xxxx.
CANNOT RETURN CONT.-yyndd	The controller associated with the specified device is not partitioned. The RETURN command is ignored.
CANNOT RETURN DEV.-yyndd	The specified device is not partitioned or is busy. The RETURN command is ignored.
CANNOT RUN AS BATCH JOB	The SYSCON processor cannot be run as a batch job.
*CANNOT WRITE M:MODNUM xxxx	An I/O error occurred when trying to write the M:MODNUM file. The error/abnormal code and subcode are specified by xxxx.
INSUFFICIENT PRIVILEGE	The user must have at least A0 privilege to run SYSCON.
INVALID TERMINATOR	The field terminator is either misplaced or unknown. When running SYSCON in the on-line mode, a dollar sign ($) is used to indicate the character position within the input command at which the error was found.
ITEM NOT PRIVATE PACK	Either the specified disk pack or one or more disk packs associated with the specified controller is not private. The command is ignored.
ITEM PARTITIONED	The resource specified in a PARTITION command has been successfully partitioned. This message will appear once for each device associated with a controller when partitioning of a controller is requested.
MODNUM COMMAND ABORTED	This message is output when any I/O error occurs during execution of the MODNUM command. The MODNUM command is aborted and SYSCON prompts for a new command.

Table 62. SYSCON Messages for the User's Terminal (cont.)

Message	Description
PARTITIONED ITEM RETURNED	A resource that was partitioned has been returned to the system. This message will appear once for each device associated with a controller when it is requested that a controller be returned to the system.
SYMBIONT yyndd TERMINATED	The specified symbiont device or the controller associated with the specified symbiont device has been partitioned.
UNKNOWN KEYWORD OR VALUE	A field in the preceding command is not valid. When running SYSCON in the on-line mode, a dollar sign ($) is used to indicate the end of the character string within the input command at which the error was found.
UNKNOWN PARAMETER	The parameter given on the preceding command is not valid for the command. The command is ignored. When running SYSCON in the on-line mode, a dollar sign ($) is used to indicate the end of the character string within the input command at which the error was found.

Table 63. SYSCON Messages for the Operator's Console

Message	Description
CONT. PART. FOR-yyndd	The controller for the specified device has been partitioned. All devices associated with this controller are also partitioned.
CONT. RET. FOR-yyndd	The controller for the specified device has been returned to the system. All devices associated with this controller that are not busy have also been returned.
DEV. PART.-yyndd	The specified nonsymbiont device has been partitioned.
DEV. RET.-yyndd	The specified device has been returned to the system.
PACK yyndd PARTITIONED DIAL ndd NOT AVAILABLE	The private disk pack spindle specified by yyndd has been partitioned. The dial number for the device is specified by ndd.
SYMB. DEV. TERMINATED-yyndd	The specified symbiont device has been terminated because it has been partitioned.
TAPE yyndd PARTITIONED DIAL ndd NOT AVAILABLE	The tape drive specified by yyndd has been partitioned. The dial number for the device is specified by ndd.

14. ON-LINE PERIPHERAL DIAGNOSTIC FACILITIES

INTRODUCTION

This chapter describes the system facilities that are designed for use by Xerox in the development of peripheral hardware diagnostic programs. The system procedures and the diagnostic DCB described in this chapter should never be used in any user-written programs. Their description is included in this manual only for completeness of documentation. Any program that uses them may seriously affect the operation and integrity of the system.

The facilities described in this chapter are used in the following types of Xerox processors:

- Functional tests for peripheral devices that isolate hardware problems to the lowest possible level.
- Exercisers that verify that the peripherals are operating correctly.
- Preventive maintenance tests that reduce the amount of time that peripherals are down for repair.

These tests and exercisers may be run at an on-line terminal while the CP-V system is in normal operation.

The facilities described in this chapter include one assembler directive, a special diagnostic DCB, and eight system procedures. The assembler directive allows the user to specify that a control section is to begin at a page boundary. The diagnostic DCB is a data area that allows the user to issue his own I/O commands. The eight system procedures perform the following functions:

Procedure	Function
M:DDCB	Generates a diagnostic data control block.
M:DOPEN	Opens the device associated with the diagnostic DCB for diagnostic purposes.
M:DCLOSE	Terminates and inhibits all I/O associated with the diagnostic DCB.
M:BLIST	Converts the user's virtual command list into a physical command list and stores the result in the diagnostic DCB.
M:SIO	Initiates the user's I/O. The commands for the I/O are stored in the diagnostic DCB.
M:LOCK	Either locks the user in core or resumes normal swapping for the user.
M:MAP	Converts a specified virtual address to a physical address or a specified physical address to a virtual address.
M:DMOD#	Obtains the controller model number, the device model number, and the type mnemonic associated with a given device address.

RESTRICTIONS

For both security and system performance reasons, there are certain restrictions on the use of the facilities described in this chapter. These restrictions are:

1. The user must have at least A0 privilege to use the procedures listed above (except for M:DDCB which generates non-executable code).
2. The procedures listed above do not reside in SYSTEM BPM. Instead, they reside in SYSTEM DIAG along with two other system procedures – M:DPART and M:DRET. (These two procedures are described in the SYSCON chapter because they are used by SYSCON.)
3. Six of the diagnostic procedure CALs cannot be processed by the system without approval from the system manager. (The exception is the M:DMOD# procedure CAL. Also, M:DDCB does not generate a CAL.) This approval is transmitted to the monitor via the key-in

   !DIAG id

   where id is the diagnostician's user ID. When a diagnostic procedure CAL is encountered, a test is made to ascertain if the user's ID is the same as the ID specified on the last DIAG key-in. If they are identical, processing continues; if not, processing is terminated. This guarantees that one and only one user may use these CALs at a given time and also provides a level of security involving verbal communication between the potential user and the computer operator.

PSECT DIRECTIVE

The PSECT directive specifies that the control section which follows is to begin on a page boundary. This directive allows diagnosticians to ensure that such things as the Diagnostic DCB and buffers start on page boundaries. The PSECT directive is described in detail in the Meta-Symbol/LN, OPS Reference Manual, 90 09 52.

SYSTEM PROCEDURES

Monitor procedures enable the user's symbolic Meta-Symbol program to request a variety of monitor functions. The on-line diagnostic procedures described in this chapter have the same general format as those described in the CP-V/BP Reference Manual, 90 17 64.

When using Meta-Symbol, the monitor diagnostic procedure library is invoked via the directive

SYSTEM DIAG

This directive defines all of the monitor procedures. The Sigma computer instruction set is invoked by the directive

SYSTEM SIG7[F][D][P]

where F specifies the floating-point option, D specifies the decimal option, and P specifies privilege instructions.

Thus, both the SYSTEM DIAG and the SYSTEM SIG7 directives should be used. The SYSTEM BPM directive should also be used if any of the procedures described in the CP-V/BP Reference Manual, 90 17 64, are used in the program.

CREATE DIAGNOSTIC DATA CONTROL BLOCK

M:DDCB The diagnostic data control block (DDCB) procedure generates a data area in the user's program that is accessible by the user. This data area must be given the label F:DIAG.

The F:DIAG DDCB must be used when the diagnostician is going to perform his own I/O through use of the diagnostic procedures described in this chapter. In addition to containing standard types of DCB information, the F:DIAG DDCB also contains the user's I/O command list (after the command list has been processed by the M:BLIST procedure described shortly). The F:DIAG DDCB format is described in detail at the end of the chapter. Because the DDCB has its own format, the only CALs that may be issued to the DDCB are the diagnostic CALs.

The M:DDCB procedure call is of the form

F:DIAG M:DDCB (DEVICE,name),(CLIST,n)[,(option)]...

where

F:DIAG is a label and must previously have been declared a dummy section via a directive of the form

F:DIAG DSECT 1

DEVICE,name specifies the device that is to be associated with the DDCB. Name may be specified in one of the following forms:

1. A device type in quotes (e.g., 'CR', 'LP').
2. An operational label in quotes (e.g., 'LO','EO').
3. The physical address of the device expressed in hexadecimal (e.g., X'A80', X'C02').

CLIST,n specifies that n words are to be reserved for the user's command list. The maximum value that can be specified for n is 24.

The options are:

SN[,{n | 'serial number'}] specifies one of the following:

1. The number of words (n) to be reserved for serial numbers. The serial numbers will be inserted into the DDCB when the DDCB is opened (M:DOPEN). The maximum value that can be specified for n is 12.
2. The serial number of the volume to be used for file input or output. The serial number may be from one to four alphanumeric characters.

If the SN option is not specified in M:DDCB, then it cannot be specified in M:DOPEN.

ABN,address specifies the symbolic address of a user's routine that is to be used to analyze any abnormal conditions associated with the creation of the F:DIAG DDCB.

The CLIST and SN options produce variable-length parameters which follow the fixed-length parameters in the Diagnostic DCB. Each variable length parameter entry is preceded by a control word of the following form:

Byte 0 is the code number (X'07' for SN; X'12' for CLIST).

Byte 1 is the code for entry position (X'00' means more parameter entries to follow; X'01' means last parameter entry).

Byte 2 is the number of significant data words in the parameter entry.

Byte 3 is the total number of words reserved for the entry, not including the control word (i.e., maximum entry length).

Special Note:

After generating the F:DIAG DDCB, Meta-Symbol resumes assembly in the control or dummy section that was in effect when the M:DDCB procedure reference line was encountered. In order to prevent the statements following the M:DDCB procedure reference line from being assembled in the same section as the DDCB, one of the following is recommended:

1. The control section directive preceding an M:DDCB reference line should be a CSECT, and the DSECT associated with an M:DDCB should precede the CSECT.

2. The statement immediately following an M:DDCB procedure reference line should be either a CSECT or a USECT referencing a prior CSECT.

OPEN DIAGNOSTIC DATA CONTROL BLOCK

M:DOPEN The monitor Diagnostic OPEN routine opens the device specified in the F:DIAG DDCB for diagnostic purposes. The DDCB will not be opened if the information in the DDCB is inaccurate, insufficient, or contradictory. In such case, the resulting abnormal or error code is returned in byte 0 of register 10. If the M:DOPEN is made with no options specified, the existing parameters in the DDCB are used. If the DDCB is already open when the DOPEN routine is called, an abnormal condition is signaled. If the DDCB is not open when the DOPEN routine is called, the DDCB is reinitialized according to the parameters specified in the M:DOPEN procedure call.

The ability to access partitioned devices is implicit in the M:DOPEN call.

The M:DOPEN procedure call is of the form

M:DOPEN [*]dcb name, (DEVICE, name), (STATUS, [*]address)[, (option)]...

where

[*]dcb name specifies the name of the DDCB (F:DIAG).

DEVICE, name specifies the device that is to be associated with the DDCB. Name may be specified in one of the following forms:

1. A device type in quotes (e.g., 'CR', 'LP').
2. An operational label in quotes (e.g., 'LO', 'EO').
3. The physical address of the device expressed in hexadecimal (e.g., X'A80', X'C02').

STATUS, [*]address specifies the address of the user's data are where the I/O status is to be stored. The status that is returned is in the same format as for the Error Log (see Chapter 12).

The options are:

SN, 'serial number'[, 'serial number']... specifies the serial number(s) of the volume(s) that are to be used for file input or output. The serial number may be from one to four alphanumeric characters. A request for the volume(s) will be sent to the operator's console. Obtaining the device requires a key-in response from the operator.

NOERR specifies that records of errors from this device are to be suppressed from the Error Log. However, the user has the option of writing records to the Error Log himself.

ABN, address specifies the symbolic address of a user's routine that is to be used to analyze any abnormal conditions associated with opening the F:DIAG DDCB.

CHAN specifies that the controller is to be reserved for use by this diagnostic program. A controller may be reserved only if it is partitioned.

Calls generated by the M:DOPEN procedure have the form:

CAL1, 6 fpt

where fpt points to word 0 of the FPT shown below.

word 0

0	1–7	8–15	16–31
*	X'05'	0———0	DCB address

word 1

0	1	2	3–14	15	16	17	18	19	20–31
P1	P2	P3	0———0	f1	f2	f3	f4	f5	0———0

word 2 - device code (P1)

0	1–15	16–31
*	0———0	Device address

device code (P1) alternate form

0	1–15	16–31
*	0———0	TEXT oplabel

where TEXT oplabel is an operational label in TEXT format.

word 3 - STATUS (P2)

0	1–14	15–31
*	0———0	Status address

option ABN (P3)

0	1–14	15–31
*	0———0	Abnormal address

Flags f_1 through f_5 in word 1 of the FPT have the significance indicated below (when f_i=1).

Flag	Significance
f_1	NOERR was specified. Error records are to be suppressed from the Error Log.
f_2	CHAN was specified. The channel is to be reserved.
f_3	SN was specified. Serial numbers are present in the FPT (in the format described below).
f_4	An operational label was specified. Word 2 of the FPT has the alternate form.
f_5	Reserved for future use.

Entries for the variable-length parameter serial number follow those for the fixed-length parameters depicted above. The format for this variable length parameter is identical to that in the F:DIAG DDCB. The variable length parameter entry is preceded by a control word of the form:

Byte 0 – Code number (X'07') identifying the variable-length parameter.

Byte 1 – Code for entry position (X'00' means more parameter entries to follow; X'01' means last parameter entry).

Byte 2 – Number of significant data words in the parameter entry.

Byte 3 – Total number of words reserved for the entry, not including the control word (i.e., maximum entry length).

If the user does not have at least A0 privilege, the return is to CAL+1 with CC1 set.

CLOSE DIAGNOSTIC DATA CONTROL BLOCK

M:DCLOSE The Diagnostic CLOSE routine terminates and inhibits I/O through the F:DIAG DDCB. I/O cannot be performed through the DDCB until it is opened again. M:DCLOSE allows the user to specify whether or not the device is down (partitioned).

The M:DCLOSE procedure call is of the form

M:DCLOSE [*]dcb name[,(PART | RETURN | SAME)]

where

[*]dcb name specifies the name of the DDCB (F:DIAG).

PART specifies that the device associated with the DDCB is to be partitioned from the system resources.

RETURN specifies that the device associated with the DDCB is to be returned to the system resources.

SAME specifies that the device associated with the DDCB is to remain in the same status (partitioned or not partitioned). The default is SAME.

The Diagnostic CLOSE routine reports the status of the device to the operator with the following message:

DEVICE yyndd {PARTITIONED | RETURNED}

where yyndd identifies the device.

Calls generated by the M:DCLOSE procedure have the form

CAL1,6 fpt

where fpt points to word 0 of the FPT shown below.

word 0

*	X'07'	0 ——— 0	DCB address

word 1

0 ——— 0	f_1	f_2	f_3	0 ——— 0

where

f_1 specifies the PART option when set.

f_2 specifies the RETURN option when set.

f_3 specifies the SAME option when set.

If the user does not have at least A0 privilege, the return is to CAL+1 with CC1 set.

BUILD COMMAND LIST

M:BLIST The monitor BLIST routine converts the user's virtual command list into a physical command list and stores the results in the F:DIAG DDCB. The routine validates that no command crosses a page boundary and converts the device address into a DCT index. It also guarantees that the number of I/O command doublewords is less than or equal to 12. The routine locks the user in core until completion of the process.

The user may optionally request that the I/O be started. If this request is made, the monitor will not return control to the user until either the request to start I/O has been rejected, the I/O is complete, or the I/O has timed-out. The AIO, TDV, and TIO status and condition codes are returned in the user area specified by the STATUS parameter of M:DOPEN and in the exact format as for Error Log (see Chapter 12).

The M:BLIST procedure call has the form:

M:BLIST [*]dcb name, (ADR, [*]address)[, (option)]...

where

[*]dcb name specifies the F:DIAG DDCB.

ADR, [*]address specifies the address of the user's command list.

The options are:

PRI, priority specifies the priority of the I/O request as a hexadecimal number (e.g., X'F6'). X'F0' is the highest priority and X'FF' is the lowest priority.

SIO specifies that the I/O is to be started.

Calls generated by the M:BLIST procedure have the form:

CAL1, 6 fpt

where fpt points to word 0 of the FPT shown below.

word 0

*	X'09'	0——0	DCB address
0	1–7	8–14	15–31

word 1

P_1	P_2	0——0	f_1	0——0
0	1	2–18	19	20–31

word 2 - ADR (P1)

*	0——0	Command list address
0	1–14	15–31

option PRI (P2)

*	0——0	Priority
0	1–14	15–31

where f_1 is set to one if SIO was specified. Otherwise, it is set to zero.

If incorrect or conflicting information exists, the abnormal address specified in the F:DIAG DDCB will be used if it has been specified. If the user has insufficient privilege, the return is to CAL+1 with CC1 set.

START I/O

M:SIO The start I/O procedure call initiates the diagnostic I/O specified in the F:DIAG DDCB. After an SIO, the monitor will not return control to the user until either the call has been rejected, the I/O is complete, or the I/O has timed-out. The AIO, TDV, and TIO status and condition codes are returned in the user area specified by the STATUS parameter of M:DOPEN and in the exact format as for Error Log (see Chapter 12).

The M:SIO procedure call is of the form

M:SIO [*]dcb name

where [*]dcb name specifies the F:DIAG DDCB.

Calls generated by the M:SIO procedure have the form

CAL1, 6 fpt

where fpt points to word 0 of the FPT shown below.

word 0

*	X'03'	0——0	DCB address
0	1–7	8–14	15–31

If there is no command list in the DDCB or the command list has been destroyed by a swap, an abnormal condition results. If the user does not have at least A0 privilege, the return is to CAL+1.

LOCK IN CORE

M:LOCK The LOCK routine either locks the user in core or resumes normal swapping for the user.

The M:LOCK procedure call is of the form

M:LOCK (YES / NO)

where

YES specifies that the user is to be locked in core.

NO specifies that normal system swapping is to resume for the user.

Once a user is locked in core, his size may not change. Therefore, the following services are not allowed:

1. Memory management CALs.
2. M:SEGLOAD, M:LINK, and M:LDTRC procedure calls.
3. Associate and disassociate CALs.
4. Get page CALs.

Calls generated by the M:LOCK procedure have the form

CAL1, 6 fpt

where fpt points to word 0 of the FPT shown below.

word 0

X'04'	0——0
0–7	8–31

word 1

0——0	f_1	0——0
0–18	19	20–31

where f_1 in word 1 specifies that LOCK in core has been requested (f_1=1) or that the LOCK is to be released (f_1=0).

If the user's privilege level is not at least A0, the return is to CAL+1 with CC1 set.

CONVERT ADDRESS

M:MAP The M:MAP procedure converts a specified virtual address to a physical address or a specified physical address to a virtual address. The converted address is stored in general register 8. The M:MAP procedure call has the form

M:MAP $\begin{pmatrix}\text{VTP}\\\text{PTV}\end{pmatrix}$, (ADR, [*]address)

where

VTP specifies virtual to physical address conversion.

PTV specifies physical to virtual address conversion.

ADR, [*]address specifies the location of the address to be converted.

M:MAP should be used with M:LOCK since the address returned via M:MAP may not be valid if a swap occurred. The user may ascertain whether or not a swap has occurred by inspecting the J:NRS field of the JIT. The J:NRS field contains the swap count.

Calls generated by the M:MAP procedure have the form

CAL, 1 fpt

where fpt points to word 0 of the FPT shown below.

word 0

0	1–7	8–15	16–31
*	X'02'	0——0	Address

word 1

0–19	20	21–31
0——0	f	0——0

where f indicates virtual to physical address conversion (f=0) or physical to virtual address conversion (f=1).

If the user's privilege level is not at least A0, the return is to CAL+1 with CC1 set.

OBTAIN MODEL NUMBERS AND TYPE MNEMONICS

M:DMOD# The DMOD# routine obtains the controller model number, the device model number, and the type mnemonic associated with a given device address. The format of the procedure call is:

M:DMOD# [*]device address

where device address has the form ndd in which n specifies the IOP unit address (the number associated with the IOP letter; see Table 60 in the previous chapter) and dd specifies the device number (see Table 61 in the previous chapter).

Example:

M:DMOD# 20F

The procedure verifies that such an address exists. If no such device address exists, CC1 is set to one. However, if the device address is valid, CC1 is set to zero and the following registers are set:

R8 (SR1) is the device model number in hexadecimal (e.g., X'00007122').

R9 (SR2) is the controller number in hexadecimal (e.g., X'00007120').

R10 (SR3) is the type mnemonic in EBCDIC (e.g., X'0000C3D9' for CR).

In either case, the return is to CAL+1.

Calls generated by the M:DMOD# procedure have the form

CAL, 6 fpt

where fpt points to word 0 of the FPT shown below.

word 0

0	1–7	8–31
*	X'0A'	Device address

word 1

0	1–31
1	0——0

ABNORMAL CODES AND MESSAGES

The codes and messages for abnormal conditions that can occur when using the on-line diagnostics facilities are listed in Table 64. (The messages reside in the system error message file, ERRMSG.)

F:DIAG DDCB

The format for the F:DIAG DDCB is given in Figure 82. Following each format, the parameter fields of the DDCB are described in alphabetical order by their mnemonic. All referenced addresses have word resolution.

Table 64. On-Line Diagnostics Abnormal Messages

Abnormal Code	Subcode	Meaning of Code
09	00	The user privilege level was less than A0.
09	01	The device referenced in the F:DIAG DCB is a nonexistent device.
09	02	The device referenced in the F:DIAG DCB is currently in use.
09	03	The device referenced in the F:DIAG DCB is currently in use by a symbiont.
09	04	The F:DIAG DCB does not contain a command list.
09	05	The command list was destroyed by a swap.
09	06	There are more than 12 I/O command doublewords (IOCDs).
09	07	The I/O command list is invalid.
09	08	An address is invalid.
09	09	A buffer crosses a page boundary.
09	0A	The user's ID does not match the ID specified on the last operator DIAG key-in.
09	0B	The amount of core is not sufficient to allow the diagnostic program to lock itself in core.
09	0C	The requested controller is not partitioned.

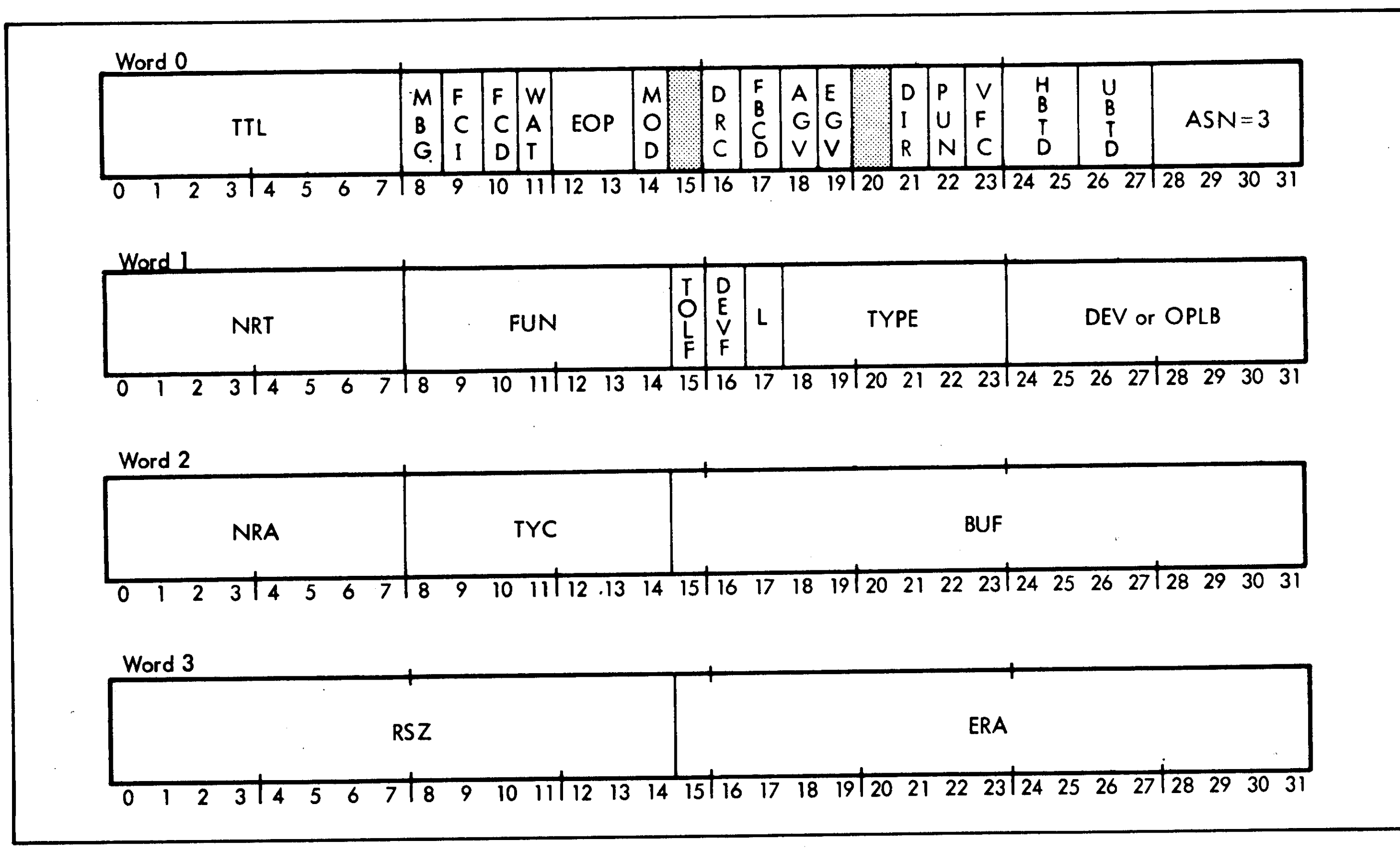


Figure 82. Format of F:DIAG DDCB

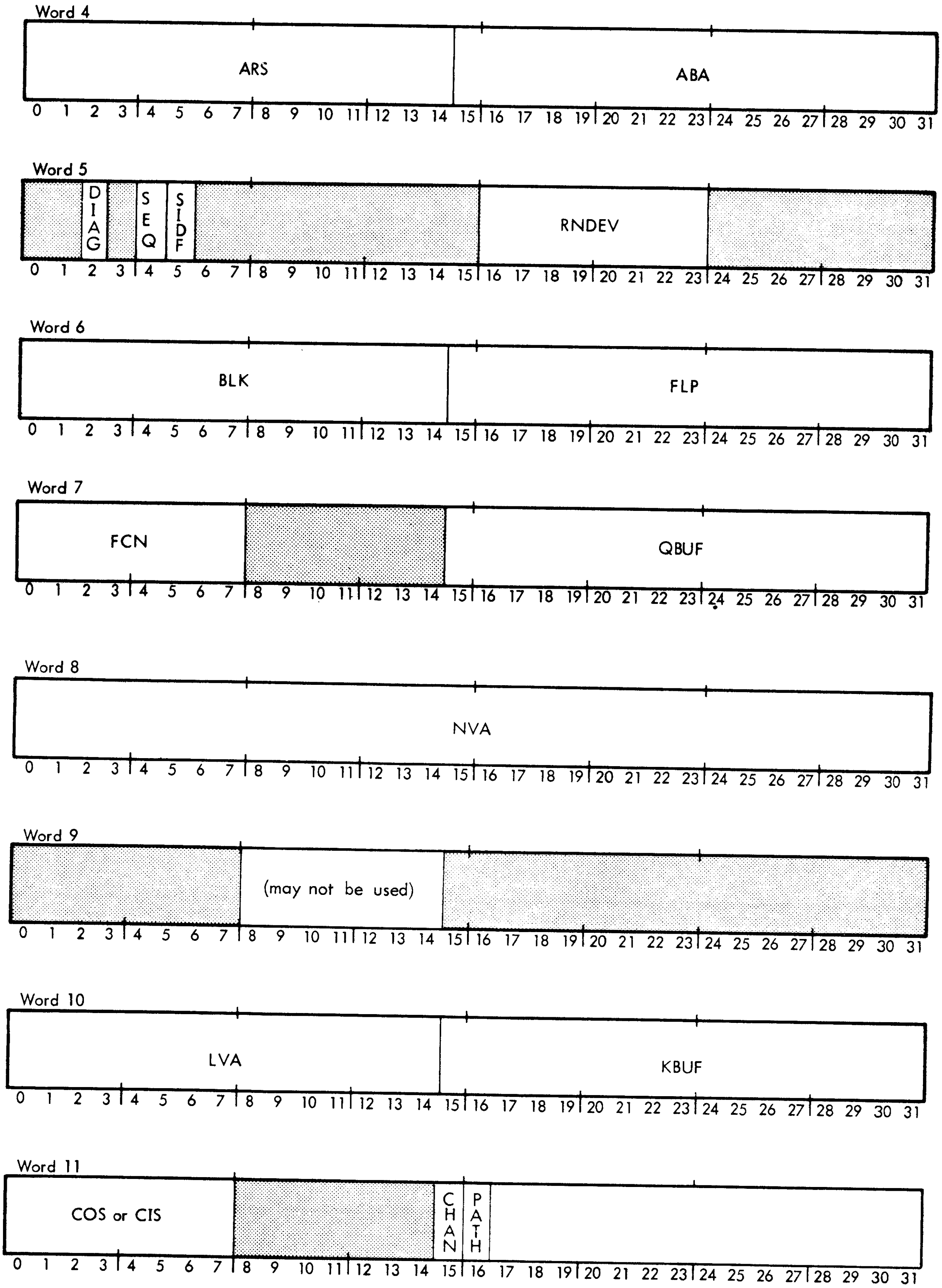


Figure 82. Format of F:DIAG DDCB (cont.)

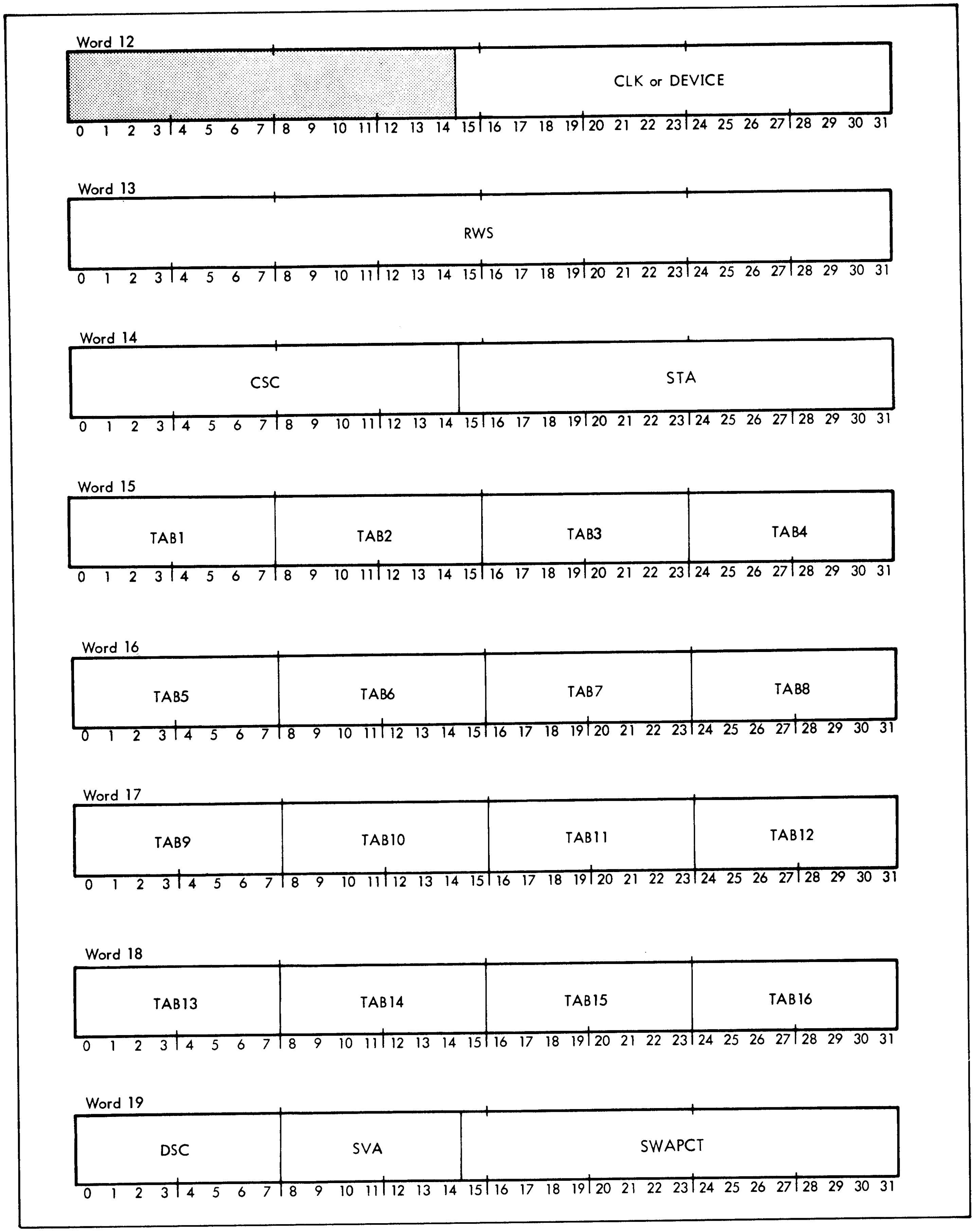


Figure 82. Format of F:DIAG DDCB (cont.)

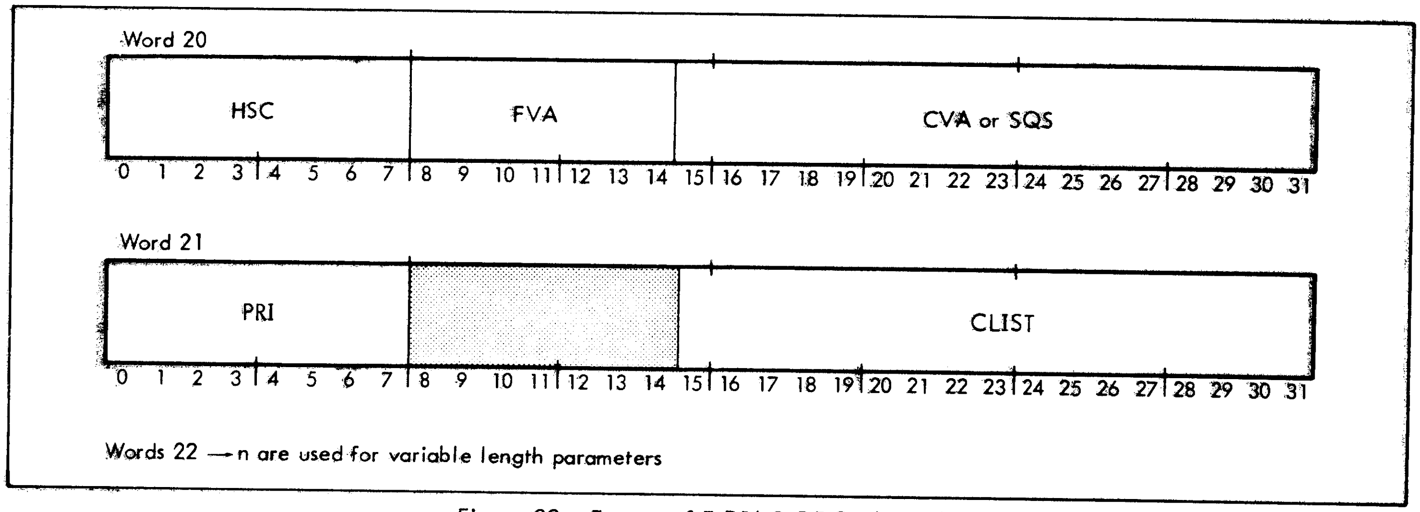


Figure 82. Format of F:DIAG DDCB (cont.)

FIELD	DESCRIPTION	WORD
ABA	Contains the address of the user's routine that will handle abnormal conditions resulting from insufficient or conflicting information. (The monitor returns to ABA in the FPT if the abnormal condition is the result of a device abnormality.)	4
AGV	Is the abnormal given flag and indicates whether or not an end-of-file completion code has been returned to the user because a control command was encountered when reading from the C device (0 = no, 1 = yes).	0
ARS	Contains the actual number of data bytes transferred to or from the user in the I/O operation.	4
ASN	Indicates the assignment type currently in effect for the DCB (0 = null, 1 = file, 2 = Xerox labeled tape, 3 = device, X'A' = ANS labeled tape).	0
BLK	Contains the number of bytes to be transferred by the I/O routines whenever called.	6
BUF	Contains the address of the user's buffer where the data record is to be read or written.	2
CHAN	Is the controller reservation flag (0 = no, 1 = yes).	11
CIS	Contains the relative position of the serial number (in the SN list) of the magnetic tape reel used for current file input.	11
CLIST	Contains the address of the physical command list in the DCB.	21
CLK	Contains the address of the first (or only) context block in the chain of symbiont or cooperative context blocks.	12
COS	Contains the relative position of the serial number (in the SN list) of the magnetic tape reel used for current file output.	11
CSC	Indicates the number of the column at which the page count is to begin (for printer or typewriter). The most significant digit of the count will be printed in this column on the page.	14
CVA	Indicates the current value of the page count (for printer or typewriter).	20
DEV	Contains the DCT index of the device assigned to the DCB. DEV is only meaningful if DEVF is set to one.	1
DEVF	Indicates whether the DCB is assigned to a device or an operational label (0 = operational label, 1 = device).	1

FIELD	DESCRIPTION	WORD
DEVICE	Contains the EBCDIC name specified on the DEVICE option in the M:DOPEN call. This use is only transient, and the field is later overlaid by CLK.	12
DIAG	Indicates diagnostic device DCB.	5
DIR	Indicates the direction of the read operation (0 = forward, 1 = reverse).	0
DRC	Is the format control flag and indicates whether or not the monitor is to do special formatting of records on read or write operations (0 = yes, 1 = no).	0
DSC	Indicates the column number at which the output record is to begin (for a card punch, typewriter, or printer).	19
EGV	Is the event-given flag and indicates whether or not the completion code posted in the TYC field has been communicated to the user's program by M:CHECK (1 = yes, 0 = no). M:CHECK is called either directly by the user or indirectly by the monitor, depending upon the WAIT, ERR, and ABN options in the FPT.	0
EOP	Is the ending operation indicator (0 = other; e.g., rewind, 1 = read, 2 = write). Specifies the type of I/O operation currently being performed or last performed.	0
ERA	Contains the address of the user's routine that will handle error conditions resulting from insufficient or conflicting information. (The monitor returns to the ERA in the FPT if the error condition is the result of a device failure.)	3
FBCD	Is the FORTRAN BCD flag and indicates whether or not BCD is to be converted to EBCDIC on input, or EBCDIC is to be converted to BCD on output (0 = no conversion, 1 = conversion).	0
FCD	Indicates whether the DCB is opened or closed (0 = closed, 1 = opened).	0
FCI	Indicates whether the DCB has ever been closed. This flag is set when the DCB is first closed, and then is never reset (0 = DCB has never been closed; 1 = DCB has been previously opened and closed).	0
FCN	Indicates the current number of I/O operations that have been initiated but not completed for this DCB.	7
FLP	Contains the address of the variable length parameters in the DCB (called the file list-pointer).	6
FUN	Contains the file mode function (0 = null, 1 = IN, 2 = OUT, 3 = IN and OUT, 4 = INOUT, 8 = OUTIN).	1
FVA	Indicates the first line on which printing is to begin (for printer or typewriter).	20
HBTD	Is the I/O handler's byte indicator and is used whenever the I/O routines are called to specify the byte displacement within QBUF into which the data transfer is to begin.	0
HSC	Indicates the column number at which the user's page header is to begin (for printer or typewriter).	20
KBUF	Is not used for devices but because of common program logic, the field contains a meaningless address.	10
L	Indicates whether or not the user specified that the DCB was assigned to a listing-type device (0 = no, 1 = yes). This flag is used only by the FORTRAN I/O routines. The monitor automatically sets this flag when the DCB is assigned to a listing-type device (such as the line printer).	1
LVA	Indicates the number of printable lines per logical page (for printer or typewriter).	10
MBG	Is the monitor buffer-flag and indicates whether or not a 34-word output buffer has been allocated to the DCB from the monitor's buffer pool. Zero specifies that the actual I/O operation will take place directly from the user's buffer. The value one specifies that the output record will be	0

FIELD	DESCRIPTION	WORD
MBG (cont.)	transferred from the user's buffer to the monitor's buffer and that the actual I/O operation will take place using the monitor's buffer.	
MOD	Is the mode flag and indicates the device mode to be used in the I/O operation (0 = EBCDIC, 1 = binary). This flag is used only when: 1. The DCB is assigned to a card punch or 7-track magnetic tape. 2. The DCB is assigned to a card reader and DRC has been specified.	0
NRA	Indicates the number of recovery tries that may be attempted before a device error message is to be logged.	2
NRT	Indicates the number of recovery tries remaining before a device error message is to be logged.	1
NVA	Contains a counter indicating the number of records to skip on magnetic tape. It is also used as an indicator. If NVA is negative, the last operation performed was a rewind.	8
OPLB	Contains the OPLB table index of the operational label assigned to the DCB. OPLB is only meaningful if DEVF equals 0.	1
PATH	Specifies path in dual-access environment (0 = no, 1 = yes).	16
PRI	Specifies priority of I/O request.	21
PUN	Indicates whether a 7-track tape is to be read or written in the packed or unpacked mode (0 = unpacked, 1 = packed). PUN is only meaningful when MOD is set.	0
QBUF	Contains the buffer address to be used by the I/O routines whenever called.	7
RNDEV	Same as TYPE field.	5
RSZ	Indicates the default record size in bytes.	3
RWS	Indicates the requested number of bytes to be read or written from the user's buffer (BUF).	13
SEQ	Is the sequence option flag and indicates whether or not punched output is to have sequencing in columns 77-80 (0 = no, 1 = yes).	5
SIDF	Is the sequence identification flag and indicates whether or not punched output is to have sequence identification in columns 73-76 (0 = no, 1 = yes).	5
SQS	Indicates the next sequence number to be output in columns 77-80 (for punched card output).	20
STA	Contains address of user data area used to return I/O status.	14
SVA	Indicates the number of lines to be spaced between printed lines (for typewriter or printer). A zero means that SPACE was not specified; the output will be single spaced.	19
SWAPCT	Contains user's swap count at the time a diagnostic CAL is issued.	19
TAB1-16	Indicates the column numbers for the tab-stop settings (for output devices).	15-18
TOLF	If 1, bits 16-31 of DCB are TEXT OPLABEL. If 0, DEVF is meaningful.	1
TTL	Specifies the length of the DCB in words.	0

FIELD	DESCRIPTION	WORD
TYC	Indicates the type of completion of an I/O completion.	2

TYC Code	Corresponding Error/ Abnormal Code	Meaning
0	0	Normal without device I/O transfer
1	0	Normal with device I/O transfer
2	7	Lost data
3	1D	Beginning-of-tape
4	4	Beginning-of-file
5	1C	End-of-reel
6	5	End-of-data
7	6	End-of-file
8	41	Read error
9	45	Write error

FIELD	DESCRIPTION	WORD
TYPE	Contains the device type code assigned to the DCB. This field is set whether the DCB is assigned directly to a device or indirectly through an operational label.	1
UBTD	Is the byte displacement indicator, specifying at which byte in the user's buffer (BUF) the data record begins.	0
VFC	Is the vertical format control flag and indicates whether or not the first byte of the output is a format control character (0 = no, 1 = yes). This flag is used only for printer output.	0
WAT	Is the wait flag and indicates whether or not WAIT was specified in the FPT (0 = no, 1 = yes).	0

15. CP-V CONFIGURATION GUIDELINES

INTRODUCTION

To arrive at a useful machine configuration to provide for the various type of users of CP-V, the manager of each installation must evaluate his requirements and the level of service he desires. A reasonable selection of hardware can only be made with a good understanding of its intended use. An important consideration is the portions of computing time to be devoted to time-sharing and batch users. Other requirements that must be evaluated include the number of on-line users, the types of tasks performed, the size of programs, and the I/O characteristics.

The following configuration diagrams and guidelines have been assembled as an aid to configuring the CP-V system. However, the Xerox Computer Sales Manual specifications supersede this information.

CORE MEMORY REQUIREMENTS

Providing enough core memory is particularly crucial in a time-sharing system. Operating with a memory of insufficient capacity for the load that is typical for an installation is especially detrimental to system performance. It reduces the system's ability to keep core loaded with ready-to-run programs and increases the frequency and duration of times when I/O is not overlapped with computing. This applies both to on-line programs and to multiprogrammed batch programs. With enough core, the system is able to keep several programs resident at once and thus to maintain a high probability of completely overlapping compute and I/O tasks. The frequency of swaps is reduced and so is the attendant system overhead, thus releasing the CPU for execution of user tasks. On-line response is doubly affected since it is directly reduced by the number of swaps and indirectly reduced by the lowering of the CPU overhead load.

The following guidelines are suggested minimum memory requirements:

80K for 24-48 terminal lines

96K for 48-64 terminal lines

128K or more for greater than 64 terminal lines

SWAP STORAGE REQUIREMENTS

The system is designed to permit a choice between two types of devices for use as swap storage – a RAD or a disk pack.

In a system that uses RAD for swap storage, the swap storage, file storage, or symbiont storage may be allocated on 7212 or 7232 RADs. In addition, file and symbiont storage may also be allocated on disk packs. The only restriction to storage allocation is that swap storage on the primary swap device must be sufficient to fulfill system storage requirements.

System storage requirements for the primary swap device are

		Size (expressed in granules)
1.	Bootstrap, HGP area, monitor overlays	approx. 90
2.	Monitor root	approx. 64
3.	Shared processors	approx. 500
	Total	approx. 654

The total implies that the following minimum storage area must be allocated on the primary swap device:

For the 7212 with 41 granules per band: about 16 bands

For the 7232 with 6 granules per track: about 110 tracks

These storage requirements will vary with the size of the monitor and the addition of shared processors to the system.

The total swap storage required for a given system may be approximated from Figure 83. Because the 7232 is a much slower device than the 7212, only a limited user system (about 20 users) should utilize a 7232 for swap storage. The average user size shown is also the average swap size and includes about 3K of user context area (system information about the user).

A system that uses a disk pack for swap storage is designed to function well in an environment of a few (8 to 24) small (6K to 12K) users. The users should be "light load" users using processors such as EDIT, FLAG, and BASIC. The system is not intended to support an environment of "heavy load" users using processors such as Meta-Symbol, FORTRAN, and COBOL.

In this system, only one disk pack spindle may be used for system residency and swapping. It may also be used for file and symbiont storage. If a second disk pack IOP is available, head movement and access conflicts can be minimized by allocating either files or symbionts to the disk packs on the second channel. The type of I/O request expected least frequently should be allocated storage on the system disk pack.

Any of the following types of disk devices may be used as the swap device: 7242/6, 7260/1, 7270, or 7275.

FILE STORAGE REQUIREMENTS

The amount of file storage is determined by the installation's requirements. It is important in terms of performance that a system be configured with adequate RAD storage to hold all master indexes whenever possible, (typically one 7232

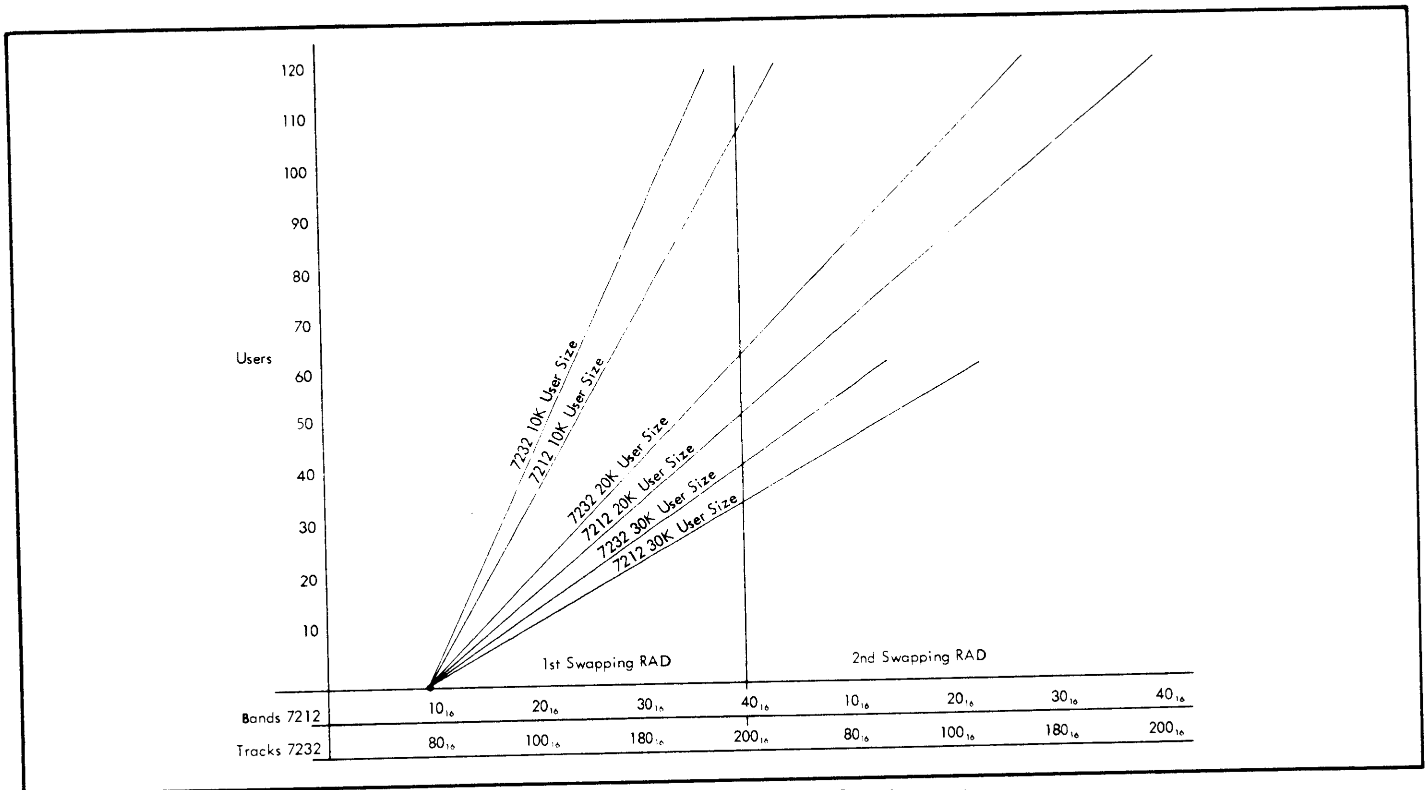


Figure 83. Estimated Swap Storage Requirements

RAD for one disk pack full of data). For each on-line user, 25,000 to 50,000 words of file storage is frequently quoted as a typical need. Thus, if 100 users have access to a 32-line system, from 2.5 million to 5 million words of file storage are needed, requiring two or three additional 7232 RADs (or equivalent disk packs).

Configurations supported by CP-V include support for any mixture of 7232, 7212, 7242, 7260, 7270, or 7275 devices for file or symbiont storage.

TERMINAL REQUIREMENTS

A variety of terminals may be used with CP-V including the Xerox 7015 Keyboard Printer, Teletype Models 33, 35, 37, and 38, IBM 2741 terminal, Tektronix Models 4010 and 4013, and Datapoint 3300. Software flexibility is present so that other terminals may be added with relatively little difficulty.

One Model 7630 Communications Controller can handle up to 64 terminal lines. More than 64 lines may be added to the system when an additional controller and interrupts are included.

PERIPHERAL EQUIPMENT REQUIREMENTS

Two on-site console keyboards are normally used for on-line monitoring and control of the system. One is the operator's console (normally a Xerox Model 7012). The other is a user type terminal (normally a Xerox Model 7015) that is connected to one COC line and is located next to the operator's console. A card reader, a line printer, and a 9-track magnetic tape unit are also required minimum equipment. It is recommended that a second magnetic tape unit and a card punch be added to the minimum equipment to provide a reasonable batch configuration.

MINIMAL CONFIGURATIONS

Figures 84, 85, 86, and 87 illustrate examples of CP-V minimal configurations. (Table 65 lists the equipment by model number, thereby supplementing Figure 84 through 89.) Although every CP-V system must have certain peripheral devices and options, the system, swap, and file storage requirements may be allocated differently to meet the expanding needs of the customer. Each example provides the essentials of a basic system, and may be easily upgraded to service more users and their demands on response time.

The examples must be considered as the recommended minimum hardware requirements and any deviation that reduces the configurations must be submitted via a field request for approval.

The first two examples (Figures 84 and 85) represent the lowest-cost minimal configuration without and with a RAD swapper, respectively. As in the other systems, the Model 7015 is used for installation control and system maintenance, while one 9-track tape unit is required for SYSGEN and backup operations. The figures show all secondary storage on

Model 7246 disk packs and the CP-V minimal memory requirement of 64K words. Because of this memory restriction, it is recommended that only a minimum number of small batch partitions be allowed to run when time-sharing operations are in progress. The suggested user load for Figure 84 is 8 to 16 terminal lines.

Figure 86 exemplifies the more typical minimum configuration. Instead of two Model 7232 RADs, user file and symbiont storage resides on disk pack, while system and swap storage remains on a single 7232. The addition of more spindles makes this configuration attractive cost-wise to the installation with a growing data base. The suggested user load for the configuration is 8 to 16 terminal lines.

The final minimal configuration (Figure 87) is designed for the installation that has a foreseeable requirement for a growing, more powerful system. The presence of a selector IOP for system and swap storage on a Model 7212 RAD will allow an easy upgrade to additional line support with the addition of more memory and more spindles. The suggested user load for the configuration is 8 to 24 terminal lines.

TYPICAL CONFIGURATIONS

Figures 88 and 89 exemplify typical CP-V configurations.

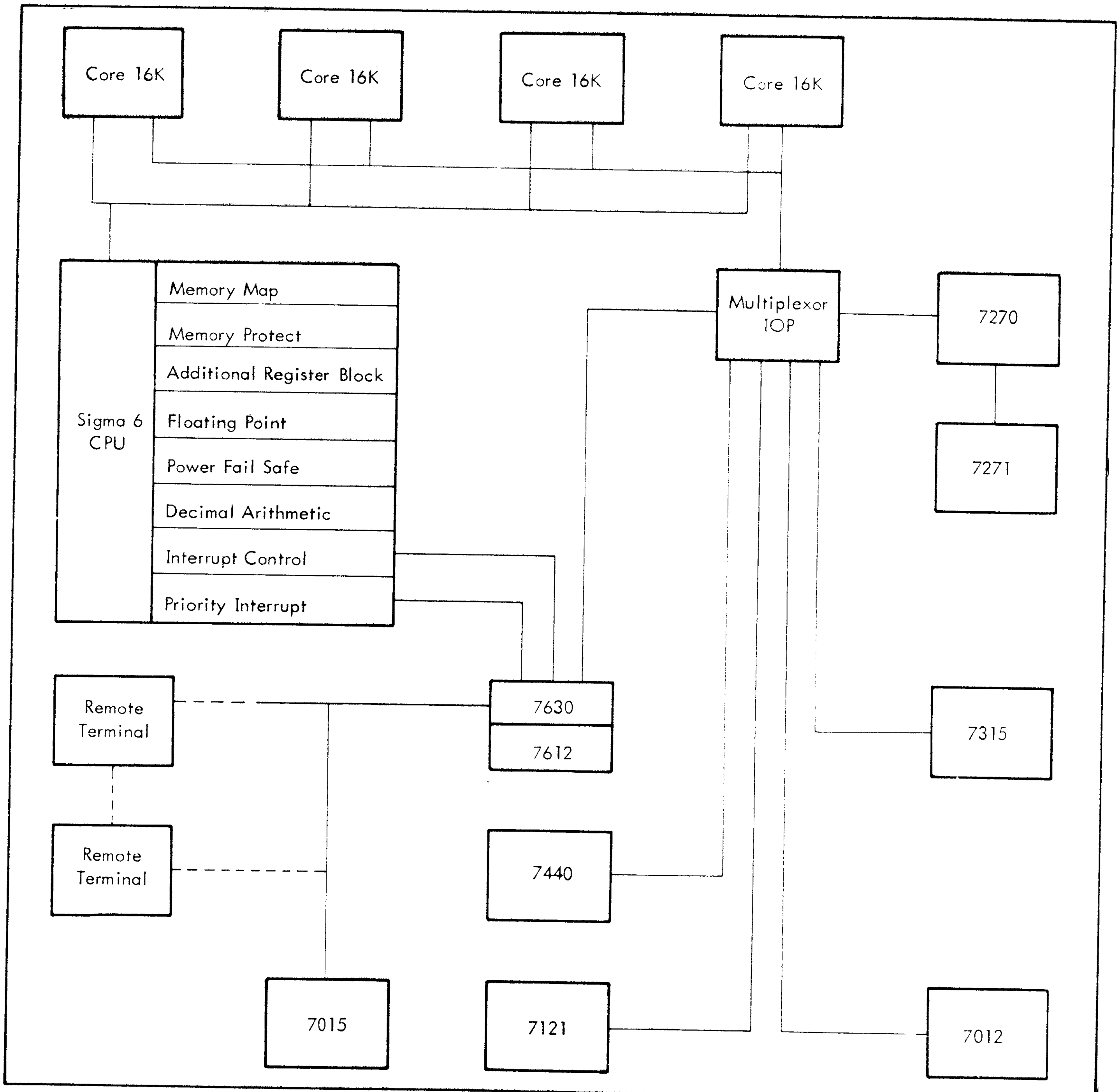


Figure 84. Lowest-Cost Minimal Configuration without a RAD Swapper

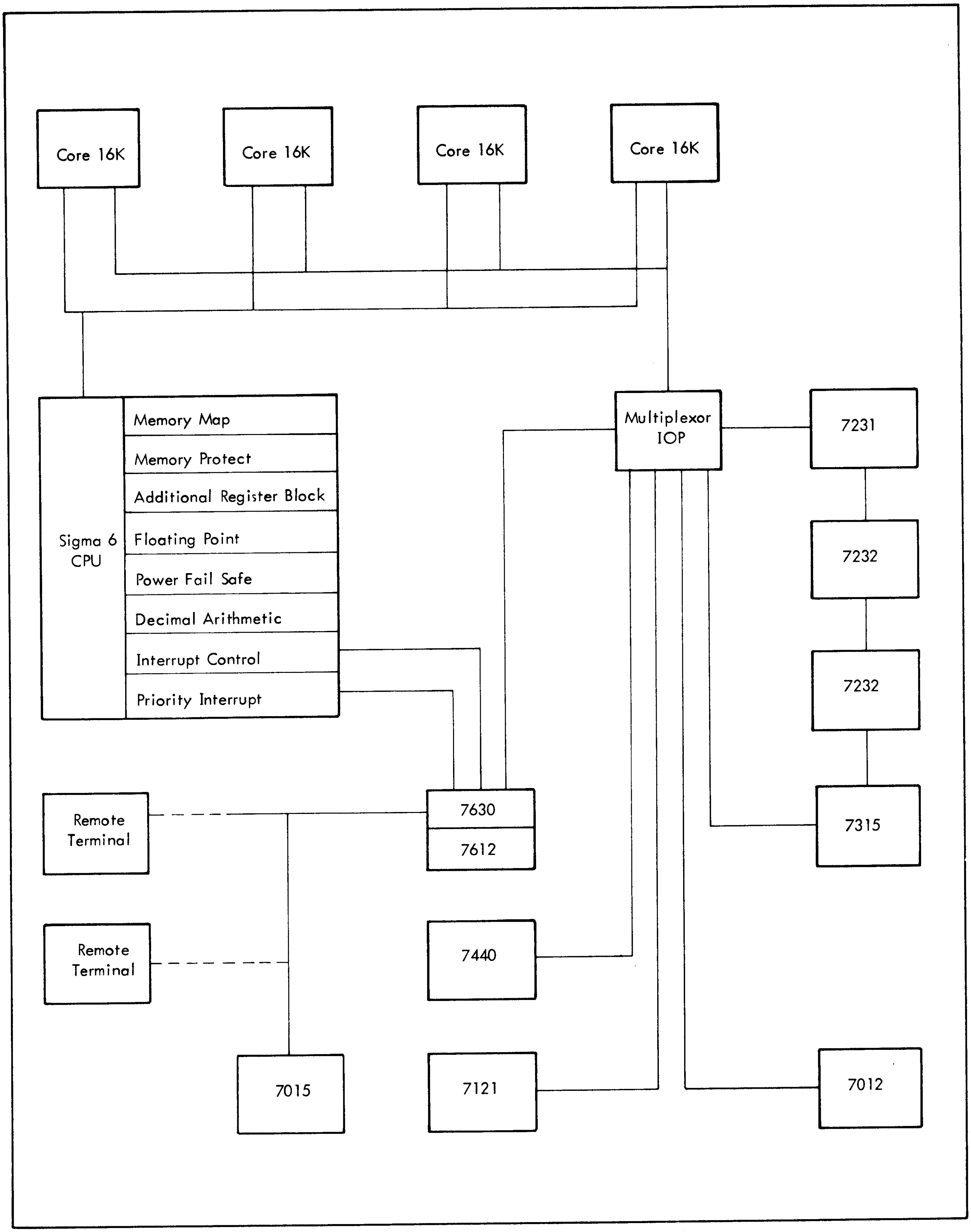


Figure 85. Lowest-Cost Minimal Configuration with a RAD Swapper

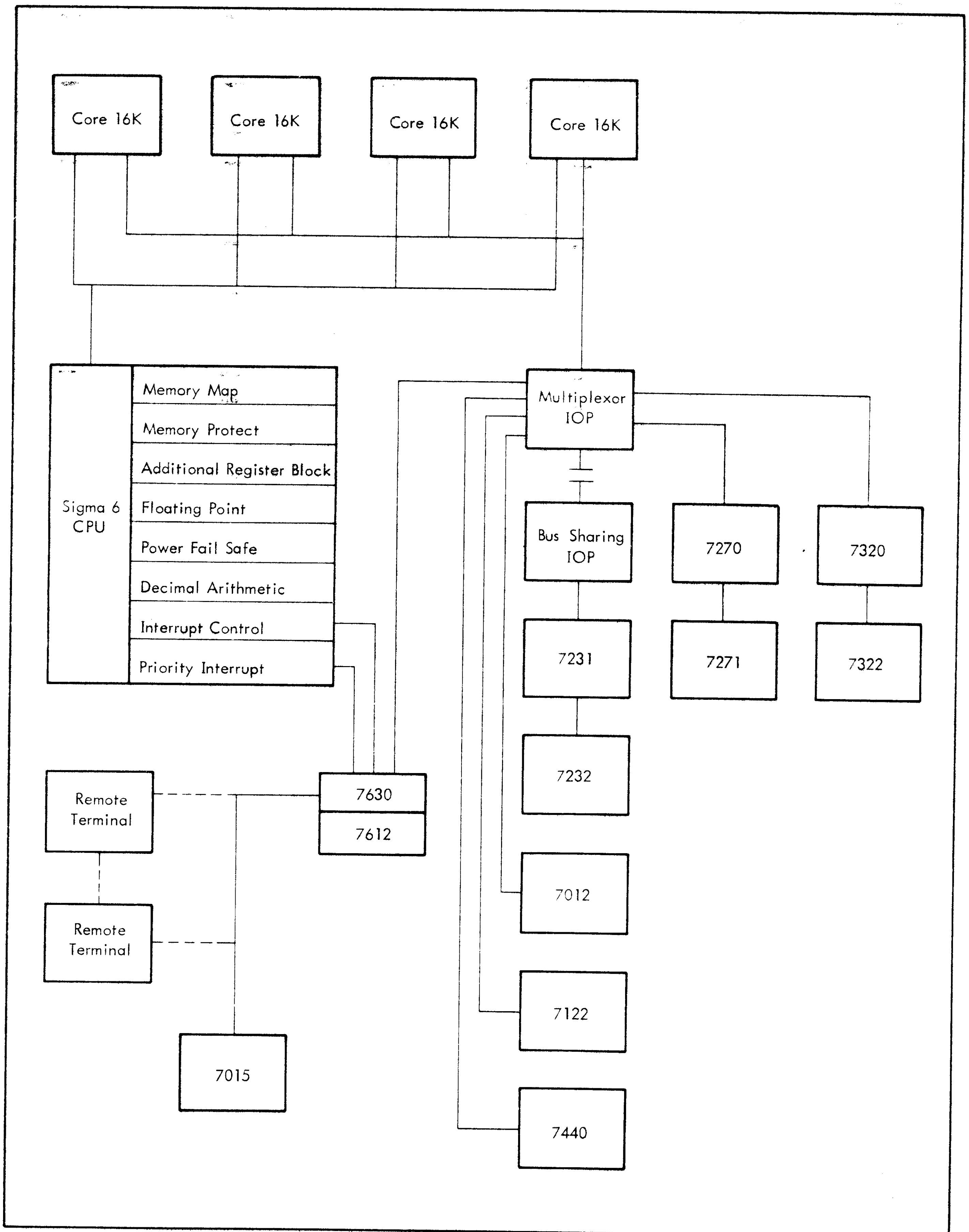


Figure 86. Typical Minimal Configuration

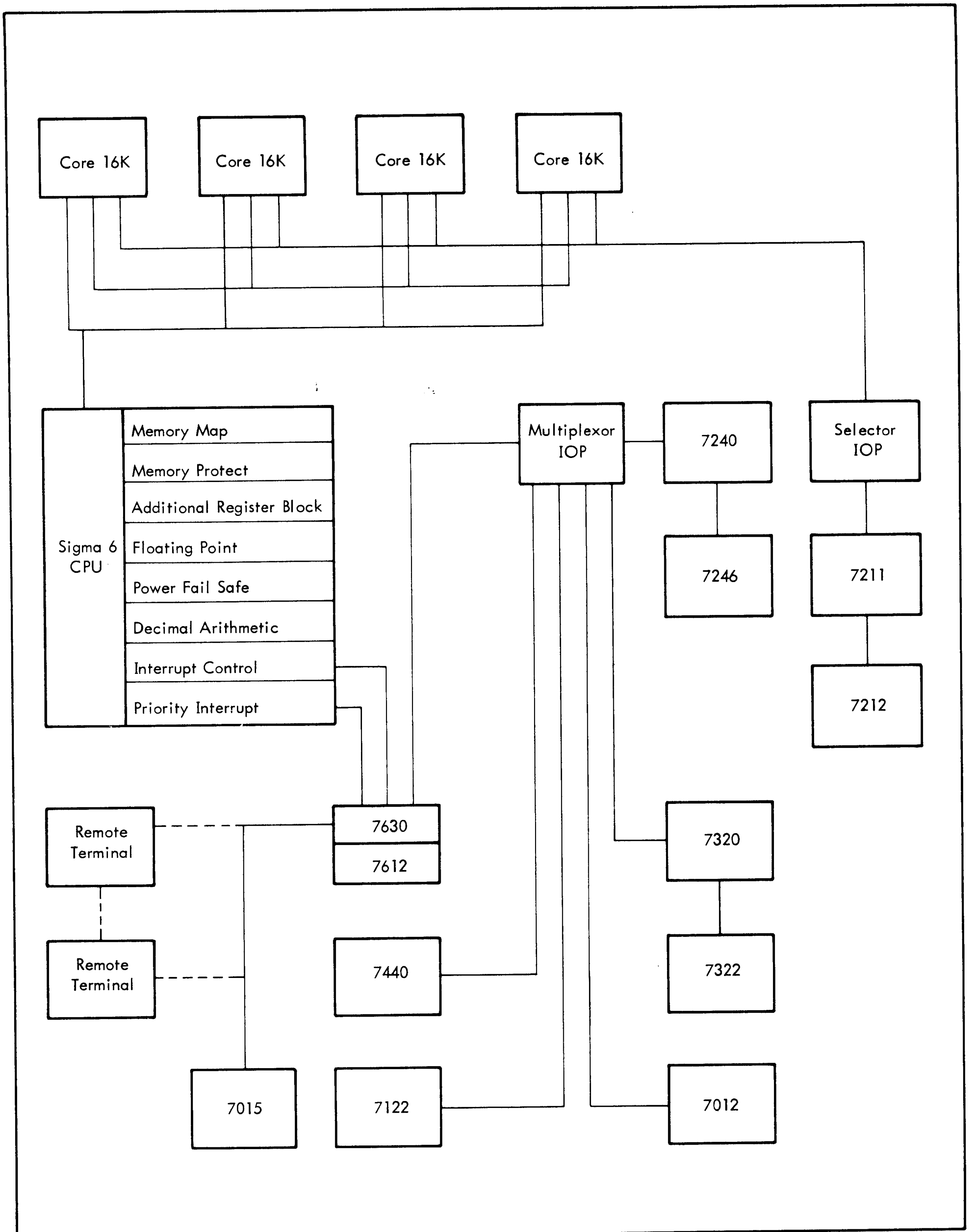


Figure 87. Typical Minimal Configuration for a Growing System Installation

Table 65. List of Equipment for CP-V Configurations

Model Number	Description
	Peripherals
7012	Keyboard printer.
7015	Keyboard printer.
7121	Card reader, 200 CPM.
7122	Card reader, 400 CPM.
7211	RAD controller.
7212	RAD storage unit.
7231	RAD controller.
7232	Extended performance RAD.
7240	Disk controller.
7242	Disk storage unit, dual spindle.
7246	Disk storage unit, single spindle.
7260	Disk controller.
7261	Disk storage unit.
7270	Disk controller.
7271	49MB disk storage unit.
7275	Disk controller.
7278	Disk storage unit.
7315	Controller plus one 60KB magnetic tape.
7320	Magnetic tape control, 800 bpi.
7322	60KB magnetic tape unit.
7330	Magnetic tape control, 1600 bpi.
7332	Tape unit; 75 ips; 1600 bpi.
7333	Tape unit; 150 ips; 1600 bpi.
7440	Line printer, 600 LPM.
7441	Line printer, 1100 LPM.
7442	Line printer, 1100 LPM.
7446	Line printer, 1500 LPM.
7450	Line printer, 225 LPM.
7612	Timing module.
7630	Communications controller plus 8 lines.
	Computer Systems
83xx	Sigma 6 systems.
84xx	Sigma 7 systems.
86xx	Sigma 9 systems.
88xx	Sigma 9, Model 2 systems.

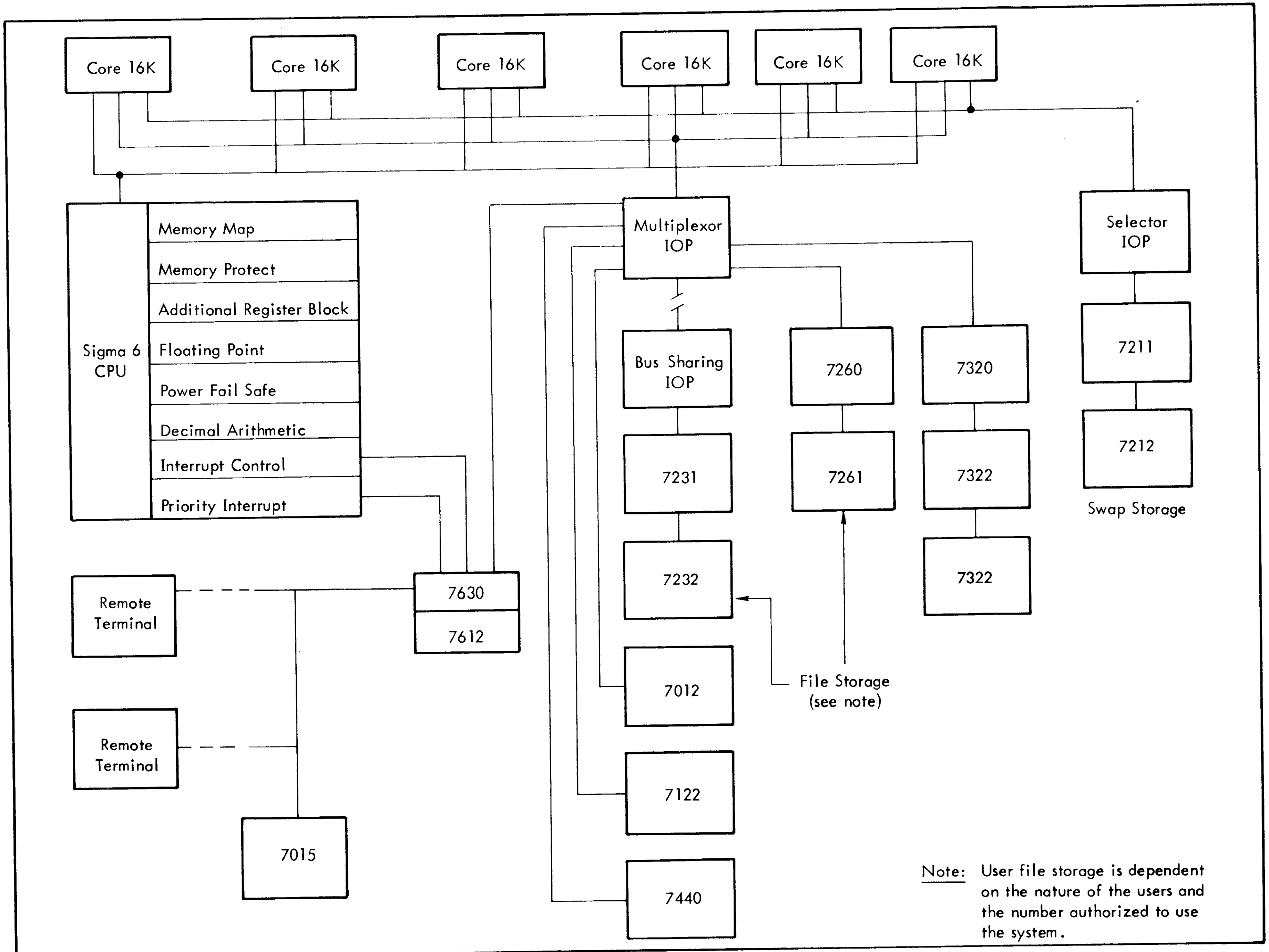


Figure 88. Typical CP-V Time-Sharing Configuration Supporting Up to 64 Lines

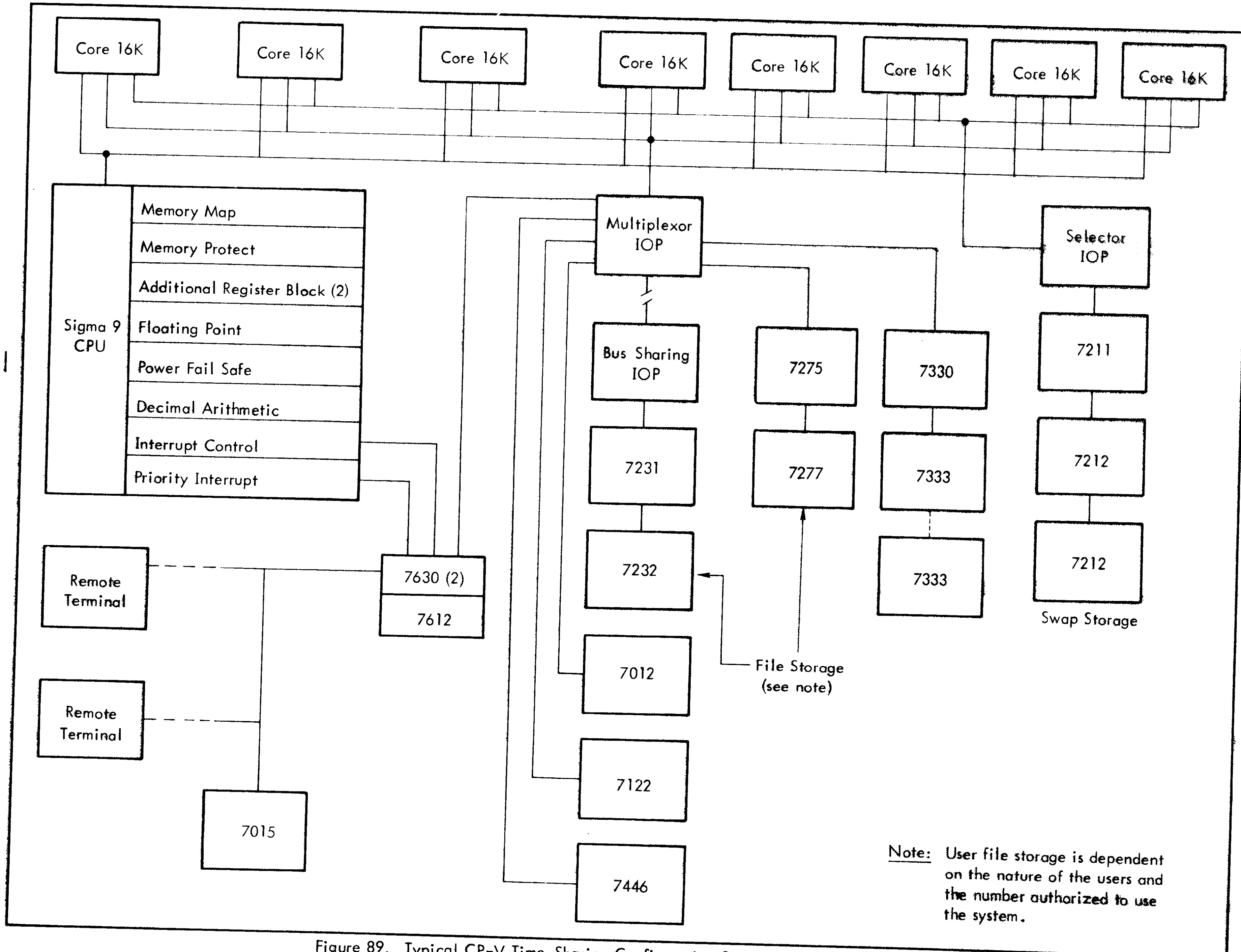


Figure 89. Typical CP-V Time-Sharing Configuration Supporting Up to 128 Lines

16. SYSTEM GENERATION

INTRODUCTION

The system generation process for the CP-V operating system is performed by service processors. These processors operate as ordinary batch jobs to collect, compile, load, and write the modules required for a system. The service processors are as follows:

Processor	Function
PCL	Selects from various sources the relevant modules for system generation.
PASS2	Compiles the required dynamic tables for the resident monitor.
LOCCT and PASS3	Stores and executes load card images (by calling the loader) to produce load modules (LMs) for the monitor and its processors.
DEF	Writes a monitor system tape that may be booted and used.

Bootstrap operations, including patching operations (PASS0), are described in Chapter 17.

Although the processors described in this chapter are collectively termed SYSGEN, they are in fact separate processors. They need not be run in any particular order or time sequence. By organizing the process properly and saving files, it is often possible to perform very short system generations to accomplish a simple task such as adding a new shared processor.

Some of the SYSGEN processors are provided for convenience. Their functions may be accomplished in some other way, if desired. For example, LOCCT and PASS3 processes may be conveniently replaced at times by simple use of load commands.

The example in Figure 90 shows how PCL and the loader are used in conjunction with DEF in the production of a new system tape and an updated copy of the monitor overlay KEYIN.

```
!JOB        :SYSGEN,1234,F
!PCL
COPYALL     LT#POX TO DC
            (Load old monitor tape into files.)
END
!LOAD       (LMN,KEYIN),(BIAS,8000),(MAP),(SL,F),;
            (NOTCB),(EF,(KEYIN),(KEYSUB),(DELPRI),;
            (DISPLAY),(IOREC),(MONSTK))
!ASSIGN     M:PO,(DEVICE,9T),(OUTSN,AP1)
            (New monitor tape.)
!DEF        CP,AP1
!FIN
```

Figure 90. Creation of a New System Tape

Commands for PASS2 fall into four groups:

1. Standard operational label assignments

   :OPLBLT specifies the default device assignment of on-line, batch, and ghost operational labels.

2. I/O control table generation and handler selection

   :CHAN groups I/O devices on a controller.

   :DEVICE gives the I/O address, handler name, device type, and general allocation (if RAD or disk pack).

   :SDEVICE defines the devices associated with symbionts.

   :LDEV defines the number and names of logical device streams.

3. Job operational limits and system control values

   :RES specifies the resources used to control multi-batch, remote, ghost, and on-line operations.

   :BLIMIT
   :OLIMIT
   :GLIMIT specifies the default and maximum limit values that apply to batch, on-line, and ghost jobs.

   :ELIMIT specifies system limit increments for exit control processing.

   :IMC specifies the initial values for system performance control parameters.

   :PART specifies the number of partitions that may be used and their resource values.

4. Monitor control table generation

   :COC specifies the number and initial values for line control tables.

   :MON specifies table sizes, CPU type, and buffer pools.

   :SPROCS lists the processors that are shared.

5. Real-time

   :FRGD specifies that the system has real-time facilities and defines the limits for real-time.

   :INTBL associates a 2-character label with an interrupt address.

6. Feature authorization

   :FAUTH specifies certain special features that may be used only by users who have been specifically authorized.

LOCCT and PASS3 provide a convenient means of retaining the load commands (LOCCT) for the monitor and all its processors and for calling the loader using these load commands (PASS3).

DEF forms a new monitor in bootable form (a PO tape) from the monitor load modules present in the current account (usually :SYSGEN). These load modules have presumably been formed by use of PCL, PASS2, PASS3, or other equivalent processes, either just prior to DEF or at some previous time. Following the formation of a bootable monitor, all load modules are written on the tape in file form. These are the target system processors.

TARGET SYSTEM PARAMETERS

Before attempting to generate a system for a specific target installation, the user must know certain things about that system. That is, he must know the desired characteristics of the system so that he can incorporate this data in the appropriate system generation control commands. The parameters that must be known beforehand are summarized below.

To define the characteristics of the target system, the user should determine the answers to the following questions:

1. Is the default set of monitor operational labels and the standard assignments for those labels adequate for the target system? If not, what standard labels and standard assignments are required?

2. What peripheral devices are needed and which of these should share the same channel controller? Which devices should be allocatable resources? Which devices will be partitionable?

3. What I/O handlers are to be used in the target system? If special handlers are needed, what are the names of the primary and secondary entry locations?

4. Are there any nonstandard devices in the target system? If so, are they listing, tape, or disk devices? What is their special handler entry points? What length is required for their command list?

5. What resources (e.g., 9-track tapes, private disk packs) are available for the system? How are these resources to be allocated among batch, on-line, and ghost jobs?

6. How many tracks are available on the RAD and/or disk pack to be used in the target system? How many sectors are there on each track, and how many words per sector? How many tracks are needed for symbiont queue storage, permanent file storage, permanent system storage, swapping storage? Which model RADs and/or disk packs are to be used?

7. How many logical device streams are required? Will the default set be sufficient?

8. How many jobs may be present at any one time in symbiont input and output job queues?

9. How large is the core memory to be used by the target system?

10. How many I/O operations may be queued at any one time?

11. How many files may be open at any one time?

12. How many DCBs may be open at any one time?

13. What is the address of the first unused, even-numbered interrupt available to the target system for the origin of the generated monitor?

14. How many words of core storage should be reserved for patching the monitor?

15. Are the standard defaults and maximums for the LIMIT parameters adequate for the target system? If not, what limits should be established?

16. What user programs, processors, or other program elements are to be established as standard systems? What load structures are to be defined for these standard systems, and what names are to be used in referencing their LOCCT tables?

17. In addition to any standard system files, what other files are to be included in the target system? Are all such files present on the current master system tape, or must some be obtained from an update tape or deck?

18. What patches, if any, must be made to the monitor or to system files after the target system has been booted from the generated system tape?

19. Is the system going to have real-time facilities?

20. Will the system have enqueue/dequeue facilities? How many concurrent enqueues will be allowed?

21. What are the limits for read-ahead?

22. Will the individual users need to be authorized to use certain special features or will all users be able to use the features?

SYSTEM GENERATION OVERVIEW

CP-V system generation is a multipass process by which the user can generate an operating system tailored to the requirements of a specific installation. Starting with a CP-V master system tape, the user can create a bootable system tape from which the generated operating system can be loaded into a target machine. The target machine can be any Sigma system having a hardware configuration compatible with CP-V (see Chapter 15) and may have more or less core storage than the one used to generate the system tape.

The master tape contains a bootable monitor, files of load modules (LMs) comprising the processors and other routines to be used during system generation, and a large number of element files (mostly ROMs) that constitute a data base for the system generation process. The user may patch the operating system as it is loaded into the machine from tape but cannot alter the ROMs at this time, since they are not read from the tape. When the monitor has been booted and the nonresident routines have been written to the disk, the CP-V system is fully operational.

Control commands read by the PCL processor allow the user to select files from the data base of the master system tape, to substitute updated files for these (if necessary), and to add files to the resulting revised data base that is maintained in disk storage for use in later phases of the current system generation. The DEF processor provides the option of writing a revised master system tape for use in some future system generation.

The PASS2 processor reads system generation control commands and generates disk files of load modules that establish operational labels, peripheral characteristics, logical device streams, allocatable system resources, and other installation-dependent parameters that will be used during a later phase of the current system generation.

The object modules selected with PCL must be combined in load module form before a generated system tape can be written. Also, the tree structures for any overlays must be established. A tree table for each CP-V standard processor is present in the master system tape. If the user references a standard processor (e.g., Meta-Symbol) in a control command during PCL, he need not make explicit reference to the tree table; it will be included automatically. However, tree tables for nonstandard systems must be created by the user through appropriate LOAD and TREE control commands.

After the user has created a tree table for an overlay structure, he has the option of calling the LOCCT processor to generate a permanent LOCCT file containing the tree information, so that this information need not be created anew during subsequent system generations.

If the generated system is to include CP-V standard systems or user-defined programs having associated LOCCT files of tree information, the PASS3 processor must be called to initiate the formation of load modules for such systems or programs.

The PASS3 processor reads control commands specifying which LOCCT tables are to be used to define the load structure of CP-V standard systems or user-defined programs.

The user may specify that a given LOCCT table and associated object modules are to be deleted from disk storage when the component object modules have been loaded.

The first command read by PASS3 should specify the monitor's LOCCT table (e.g., M:MON), so that the monitor will be loaded first. Any items loaded will be biased to that bias contained in the LOCCT table for the item (this bias comes from the original LOAD control command used to generate the LOCCT table being used).

Items not specified in PASS3 control commands may be loaded via LOAD, OVERLAY, and TREE commands as in ordinary batch processing, except for the monitor load module (M:MON). M:MON must be loaded via PASS3.

When all desired object modules have been converted to load module form, the DEF processor must be called to write a tape containing the generated system.

The system tape generated by the DEF processor has the same general format as the master tape used in booting the CP-V system employed in the system generation process. This format is shown in Figure 91. The method of loading the generated system into the target machine is identical to that used in booting from the master tape.

Detailed procedures for generating standard (i.e., typical) CP-V systems, and detailed descriptions of the various control commands used in system generation are presented later in this chapter.

COMMAND FORMATS

The control commands used in system generation are of two general types: monitor control commands having an "!" in position 1 (e.g., commands used to call system generation processors for execution), and system generation control commands having a ":" in position 1 (i.e., commands used to communicate optional or required parameters to system generation processors).

System generation commands that have an asterisk in position 1 are interpreted as comments. If comments are desired on a control command, the comment must be preceded by a period or a semicolon.

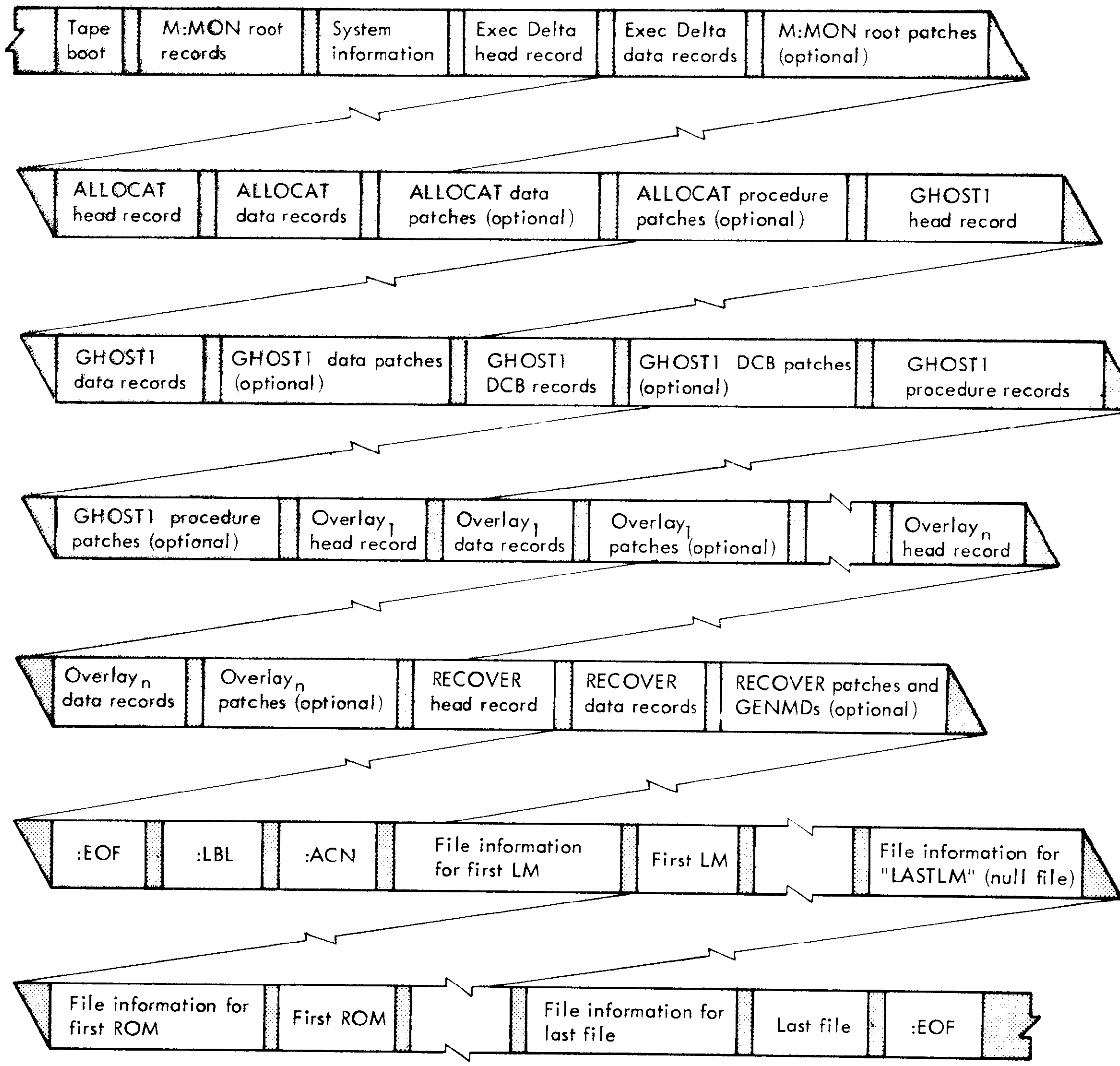
All system generation commands may contain continuation characters. A continued command is identified by a semicolon. All continued commands must contain a colon in position 1.

Since the user has considerable flexibility in setting up and performing a system generation, it is not practical to present extensive system generation examples in this manual. (Examples are provided in the release documentation which is part of the CP-V release package.) However, by observing the general considerations outlined in this chapter and the examples, the user should have little difficulty in setting up his own system generations.

Many users will find that the "cookbook" job setups given for generating standard CP-V systems are adequate for their needs, and it is likely that the requirements of most installations can be met by making relatively minor alterations to one of the standard configurations.

PERIPHERAL CONVERSION LANGUAGE (PCL)

The copy commands of the PCL processor are used to obtain the files required for a SYSGEN from the master tape and



Note:

Record sizes	The tape bootstrap is 22 words long. Patch records are 80 words long. All other records are 512 words long. The figure indicates groups of such physical records.
Head	Head portion of load module.
Data	Protection type 0 portion of load module.
DCBs	Protection type 2 portion of load module.
Procedure	Protection type 1 portion of load module.
OVERLAY$_i$	M:MON overlays (shared processor type) as described in M:SPROCS module (e.g., OPEN, CLOSE, KEYIN, DEBUG, LDLNK, MUL, LTAPE, MISOV, IODTYPR, STEOVR, RMAOV, RTNRRT, ENQ).
OVERLAY$_n$	Last M:MON overlay as described in the M:SPROCS module.
Patches	Patches are included on the tape where shown if they exist in the file assigned to the M:PATCH DCB when DEF creates the system tape. The first group of M:MON root patches follows the Exec Delta data records. Any others are placed among segment patches according to their order in the patch file. The last record of each group of patches on the tape is the first patch for the next set of segment patches. The second through the last patch for a segment follow the segment to which they will be applied. GENMD patches follow the last of any patches following the RECOVER patches.

Figure 91. Format of Master System Tape

other sources. The functions that these commands perform that are applicable to SYSGEN are

COPY copies files from tape or disk to tape or disk.

COPYALL copies every file from a tape or disk account to another tape or disk account.

COPYSTD copies files as specified in a STANDARD file.

For detailed information on the use of PCL, see the CP-V/TS Reference Manual, 90 09 07.

PASS2 PROCESSOR

The PASS2 processor generates the various tables required for the target monitor and assigns values to variables that control the system. Table sizes conform to the physical requirements of the target monitor. PASS2 reads control commands from the SI device and creates an intermediate temporary keyed file. The M:MODNUM file in the :SYS account is read through the M:EI DCB. However, this DCB can be reassigned so that the M:MODNUM file is read from some other account. The records (the : commands) of the file are then accessed in the order that PASS2 requires to build the dynamic monitor tables.

PASS2 COMMANDS

Commands recognized by PASS2 are listed below (in the order of their appearance in the chapter).

:CHAN	:BLIMIT	:MON
:DEVICE	:OLIMIT	:SPROCS
:SDEVICE	:GLIMIT	:PART
:LDEV	:ELIMIT	:FRGD
:RES	:IMC	:INTLB
:OPLBLT	:COC	:FAUTH

With the exception of the :CHAN, :DEVICE, :SDEVICE, and :COC commands, PASS2 will build dummy commands for those omitted by the user, thereby generating the various load modules with SYSGEN default values.

Any entry preceded by an asterisk in column 1 is interpreted as a comment. Comments may also be part of a control command by preceding the comment with a period or a semicolon. However, a semicolon requires a continuation card.

!PASS2 This command causes the monitor to fetch the PASS2 processor from disk and to transfer control to it. The PASS2 command has the form

!PASS2 [CP]

where CP is the monitor type and need not be specified.

:CHAN This command groups peripheral devices (see :DEVICE below) according to channel controller. All :DEVICE commands following a given :CHAN command are assumed to be a part of that channel. At least one :CHAN command must be used, and each :CHAN command must precede the :DEVICE command (or commands) to which it applies. The :CHAN command has the form

:CHAN [(DUAL, ($n_1c_1$0, $n_2c_2$0))][, (NOPART)]

where

n_1c_1 specifies the IOP/controller fields of the primary I/O address.

n_2c_2 specifies the IOP/controller fields of the secondary I/O address of the dual-access controller, e.g., (D80, E80).

Both single and dual-access devices may be included under the same logical channel. The :DEVICE cards for dual-access devices shall specify only the primary I/O address, i.e., $n_1c_1d_1$. No more than one dual-access controller may be on any given logical channel.

NOPART specifies that the controller cannot be partitioned.

:DEVICE This command specifies the name and characteristics of a system peripheral device. One :DEVICE command must be used for each device in the target system.

The :DEVICE command has the form

:DEVICE name, (MOD, [D,]dddd[, cccc])[, (option)]...

where

name specifies the device name in the form yyndd (see Appendix C) with NO, MT, and SP being invalid. For a nonstandard device type, the two position yy field is included as a standard device type and may be referenced elsewhere in PASS2 as a resource type or as a symbiont type.

MOD indicates that the device identified by name has the model number dddd and the controller has the model number cccc. The dddd and cccc fields each consist of four hexadecimal digits. (The dddd and cccc fields are verified by reading the M:MODNUM file from :SYS or from the account assigned to the M:EI DCB. See the SYSCON processor.)

D specifies that the device is either a RAD or a disk pack. In this case, dddd must be one of the following:

7202 7203 7204 7212 7232	Specify the type of RAD storage device (DC).
7242 7261 7271 7275	Specify a disk pack (DP).

The 7202, 7203, and 7204 RADs may not be used as system RADs (i.e., may not be used for file, symbiont, or swap storage). They may only be used as private devices.

{INPUT | OUTPUT | IO} specifies whether the device is to be used for input, output, or both. The default is IO. All standard device types (listed in Table 66) ignore this option.

HANDLER, name1, name2 specifies the name of the I/O handler to be used. Name1 is the primary entry (build command list and start device) and name2 is the secondary entry (handler interrupt). Neither name may exceed seven alphanumeric characters. Name1 must be the name of the object module for this particular handler. If this option is omitted, the default handler for the device type is assumed (Table 66). The names in Table 66 must be used unless the user has supplied his own handler (in the root of the monitor).

When defining a device that is listed in Table 66, no HANDLER option is needed unless the name1, name2 defaults for that device are to be changed. Conversely, when a :DEVICE command defines a device that is not in Table 66, the HANDLER option must be defined. This allows unique device addresses with identical device types to be controlled by unique handlers. The CLIST option should be used if the default value of six is not sufficient for the nonstandard device handler.

Example:

```
:DEVICE LPA02
:DEVICE LPB02, (HANDLER, PRTOUTL, PRTCU)
:DEVICE LPC02, (HANDLER, PRTIO2, PRTCU2)
```

For device LPA02, the default handler entry names will be used (see Table 66). Otherwise, the handler addresses will be as stated.

When all :DEVICE commands have been processed, PASS2 builds a record containing the names of the handlers needed for this particular target machine. This file is used by SYSGEN PASS3. The handler's element files on the SYSGEN BI tape, or those entered into the system by other means, must be given file names that correspond to their name1 entry points.

Table 66. Default Handler Entry Points

Device Type	Name 1	Name 2
TY	KBTIO	KBTCU
PR	PTAP	PTAPCU
PP	PTAP	PTAPCU
CR	CRDIN	CRDINCU
CP	CRDOUT	CRDOCU
CP Model 7165[†]	CRDOUTL	CRDOCU
LP	PRTOUT	PRTCU
DC	DISCIO	DISCCU
9T	MTAP	MTAPCU
DP	DPAK	DPAKCU
DP[††]	DISKAB	DSKABCU
ME	COC	COC
7T	7TAP	7TAPCU
RB	–[††]	–[††]
XP	OCPIO	OCPCU

[†] Handlers must be specified.

[††] The default handlers should be used. The handler entry points are dependent upon the type(s) of remote terminals specified (see below).

NOPART specifies that this device cannot be partitioned.

PAPER, size, width specifies, in hexadecimal, the number of printable lines per page (size), and the maximum number of characters per line (width).

This option applies to TYs, LPs, and RBs. The defaults are: 26_{16} and width 84_{16} (for RB, width default is 80_{16}).

The following options apply to remote processing (device RBndd). They define the characteristics of a data set controller (DSC) rather than remote peripheral devices. Defining a data set controller establishes remote processing as a feature of the particular CP-V installation and causes all the remote processing handlers and tables to be included as part of the system.

$\begin{Bmatrix} 7670 \\ 2780 \\ \text{IRBT} \end{Bmatrix}$ specifies whether the DSC is usable for Xerox 7670 RBTs, IBM 2780 RBTs, or HASP Multileaving IRBTs. IRBT and 2780 may both be specified for the same DSC, but no other combination is legal. If both 2780 and IRBT are specified for DSCs, either separately for different DSCs or together on the same DSC, either type of terminal can be connected to any DSC for which either 2780 or IRBT (or both) is specified. In this case, for example, an IRBT may be connected to a DSC which was defined for use with 2780s.

$\begin{Bmatrix} \text{FULL} \\ \text{HALF} \end{Bmatrix}$ specifies whether the DSC is full-duplex or half-duplex. The default is HALF.

A full-duplex DSC must be configured on two adjacent MIOP subchannels because of a hardware restriction. If FULL is specified and the device number (dd) is odd, the alternate address is dd-1; if dd is even, the alternate address is dd+1.

$\begin{Bmatrix} \text{RBS} \\ \text{RBX} \end{Bmatrix}$ specifies that the line is ready for use at boot-time (RBS) or that the operator must use the RBS key-in to make the line ready for use (RBX). The default is RBS.

WSN,name specifies a one- to eight-alphanumeric character workstation name that is to be automatically associated when the line is connected.

The following options may be used in conjunction with the other options to define nonstandard devices. It is assumed that all devices with the same device type have the same characteristics. These options should not be used with the standard device types (Table 66).

L specifies that the device is a listing type device.

T specifies that the device is a tape type device. If no HANDLER option is used, PASS2 will default the handler to MTAP,MTAPCU (standard 9-track handler) and will define a CLIST of eight words.

PUB specifies that the device is a public device.

COMP specifies that the records are to be compressed; that is, trailing blanks are to be stripped from records output on the device down to the minimum record size.

VFC specifies that vertical format services such as page counting, headers, and line spacing are legal on this device.

BIN specifies that binary operations are allowed on the device. The monitor will produce BCD, BIN, and EOD commands as appropriate.

R specifies that the read reverse operations are legal on this device. This option applies to tape type devices. Its use is required for 9-track tape devices if other than 9T is specified for the device type.

MXREC,value specifies, in hexadecimal, the maximum record width or size. Note that for listing devices, the PAPER option must be used to define the width because PASS2 ignores MXREC for listing devices.

MREC,value specifies, in hexadecimal, the minimum record length.

CLIST,value specifies, in hexadecimal, the number of words to be allowed in the command list table for a given device. For nonstandard devices, the default is 6.

The allocation of RAD or disk pack area may be constrained by the following options:

SIZE,value specifies, in hexadecimal, the amount of storage available to the system device. For RAD devices, the value must be expressed in number of tracks. For disk pack devices, the value must be expressed in terms of physical cylinders unless the CYLIN option is also specified, in which case the value must be expressed in terms of logical cylinders. If omitted, the value is assumed to be the sum of PSA, PER, and PFA (see below).

NSPT,value specifies, in hexadecimal, the number of disk sectors per track. If omitted, the value C_{16} is assumed.

SS,value specifies, in hexadecimal, the number of words per disk sector. If omitted, the value 100_{16} is assumed.

PER,value specifies, in hexadecimal, the amount of storage to be allocated for peripheral symbiont queue storage. For RAD devices, the value must be expressed in number of tracks. For disk pack devices, the value must be expressed in terms of physical cylinders unless the CYLIN option is also specified, in which case the value must be expressed in terms of logical cylinders. The default value is 0 (note that one minute of backup for an 800 line per minute printer uses $25,000_{10}$ words of disk scratch storage).

Note: During execution when the number of granules of PER is exhausted, the system will use PFA.

PFA,value specifies, in hexadecimal, the amount of storage to be allocated for permanent file storage, including element files. For RAD devices, the value must be expressed in number of tracks. For disk pack devices, the value must be expressed in terms of physical cylinders unless the CYLIN option is also specified, in which case the value must be expressed in terms of logical cylinders. The default is 0.

PSA, value specifies, in hexadecimal, the amount of storage to be allocated for system use as permanent system storage and swapping area. The default is 0. Swap storage may be allocated on either RAD or a disk pack. (See Chapter 15 for a guide.) For RAD devices, the value must be expressed in number of tracks. For disk pack devices, the value must be expressed in terms of physical cylinders unless the CYLIN option is also specified, in which case the value must be expressed in terms of logical cylinders.

RAD

When the system is initialized in boot-time, PSA tracks from the first PSA RAD are allocated to permanent system storage. The first PSA value should therefore be great enough to allow system residency (about 10 tracks on a 7212 RAD). Swap storage can be allocated on more than one RAD. The total PSA on all RADs should be at least equal to

(number of users) x (average user size) + (space for system)

Disk Pack

The PSA option must be used only once if it appears on a disk pack :DEVICE command. (Only one disk pack may be used for swap storage and a disk pack cannot be used for swap storage if a RAD is also being used for swap storage.)

The total PSA on a disk pack should be at least equal to

(number of users) x (number of cylinders per user) + (space for system)

The number of cylinders per user is determined by the maximum user size. System residency for a 7242/7271 requires approximately 12 cylinders. The 7261/7275 requires approximately 8 cylinders.

CYLIN, value specifies that a "logical cylinder" allocation table is to be built instead of a granule allocation table and is used for moveable head disk devices only. The value represents, in hexadecimal, the number of granules per logical cylinder and may be between 1E and FF. If a value is specified that is greater than FF, PASS2 will print an error message and substitute the default value for the given device type. If the device is nonstandard, PASS2 substitutes the value 1E. If the total number of granules on the device is not evenly divisible by the NGC value, PASS2 prints a message indicating this and ignores the remainder of the command.

If the CYLIN option is used, the SIZE and PFA options must be specified in number of logical cylinders rather than number of physical cylinders.

PASS2 will set PFA equal to SIZE if CYLIN is specified. If SIZE is set by default, it is automatically set to the appropriate value for the standard device. The PER and PSA options are ignored for logical cylinder allocation. If the CYLIN option is not specified, PASS2 assumes granule allocation.

If the value selected for CYLIN for a private disk device causes the allocation table to exceed one page, PASS2 prints an error message and aborts.

PRIVATE specifies that the file will be recognized as private and CYLINder allocation is forced. The PRIV bit is set in the allocation table, the PUB bit is reset in the Automatic Volumn Recognition (AVR) table, and PFA is set equal to SIZE ignoring the options PER, PSA, and PFA.

Table 67 identifies the fixed characteristics and the PASS2 default values of disk devices. All values are expressed in hexadecimal.

Table 67. Standard Device Type Option Default Values

OPTION	SIZE	SS	NSPT	NCYL	NTPC	CYLS	TRKS	SECS	CYLIN
7232	200	100	C	–	200	–	14	10	–
7212	40	100	52	–	40	–	17	10	–
7242	FA0	100	6	C8	14	10	8	–	1E
7261	FA0	100	B	C8	14	10	8	–	37
7271	1F40	100	6	190	14	10	8	–	30
7275	1DFC	100	B	194	13	10	8	–	34

Nonstandard Disk Devices

The following options are included in the :DEVICE command in order to define nonstandard fixed or moveable head disk devices. These options should not be used in defining standard devices since PASS2 has defaults for these values. Note that the previously defined options SS, NSPT, and SIZE should also be used for defining nonstandard disk devices.

{FIXED / MOVE} identify the disk devices as either FIXED or MOVEable head disk devices.

NCYL, value specifies, in hexadecimal, the total number of physical cylinders on the device. This parameter applies only to moveable head disk pack devices. Note the previously defined option CYLIN refers to logical or software cylinders.

NTPC, value specifies, in hexadecimal, the number of tracks per physical cylinder. For fixed head disk devices, this value represents the total number of tracks on the device.

CYLS, value specifies, in hexadecimal, the shift factor for the cylinder portion of disk addresses. This parameter applies only to moveable head disk pack devices.

TRKS, value specifies, in hexadecimal, the shift factor for the track portion of disk addresses. This parameter applies to both fixed and moveable head disk devices.

Default Tape and Disk

For compatability with previous monitor versions, two device names, MT and SP, are generated automatically and are given default device assignments.

MT is the default tape device and will be assigned according to the following rules:

1. If a 9T device has been declared on a :DEVICE command, MT is equivalent to 9T.

2. If no 9T device has been named but a 7T device has been declared, MT is equivalent to 7T.

3. If neither 9T or 7T has been declared, MT is equivalent to the first tape device (T specified on :DEVICE card) which was mentioned.

SP is the default disk device and will be assigned according to the following rules:

1. If a DP device has been declared on a :DEVICE card, SP is equivalent to DP.

2. Otherwise, DP is equivalent to the first disk pack device which was mentioned.

:SDEVICE This command is used to designate which peripheral devices are to be symbiont. The form of the command is

:SDEVICE (option)[,(option)]...

where

IN, yyndd specifies that the peripheral device yyndd is to be associated with the input symbiont. Only one yyndd may follow the keyword IN. However, multiple occurrences of the keyword may be used.

OUT, yyndd specifies that the peripheral device is to be associated with the output symbiont. Only one yyndd may follow the keyword OUT. However, multiple occurrences of the keyword may be used.

NCTL, yyndd specifies that the peripheral device is to be associated with the input symbiont and that the device will not be scanned by the input symbiont for control commands except for !EOD and !FIN commands. This means that files input through this device are not in the form of jobs but rather are symbiont input files accessed by users via the LDEV command. Only one yyndd may follow the keyword NCTL. However, multiple occurrences of the keyword may be used.

MXSTRM, value specifies, in decimal, the maximum possible number of devices (symbiont streams) at concurrently logged on remote workstations. The maximum value that may be specified is 128. Three streams are required for each Xerox 7670 RBT. IRBTs usually require more. The minimum and default for systems in which remote devices are defined is three times the number of RBTs defined. For other systems, the value is 0.

Note: RB devices are ignored if specified on the :SDEVICE command. Their existence should be accounted for in the use of the MXSTRM option.

:LDEV This command is used to define the names of logical device streams. The maximum number of logical device streams that may be defined by the command is 15. The format of the command is

:LDEV [(name, type)][,(name, type)]...

where

name specifies the two-character logical device name (e.g., L1, P1).

type specifies the default device type to be associated with the logical file (e.g., LP, CP). The device type specified must have been defined via a :DEVICE command and must also be designated as a symbiont device via the :SDEVICE command.

If the user omits the :LDEV command or no options are specified, PASS2 generates the following entries by default:

Name	Type
C1	CR
L1	LP
P1	CP

The logical device name C1 is always automatically generated with type CR and should not be defined by the user in the :LDEV command. (Because the system manager can define up to 14 logical device names, a total of 15 logical device names are possible with the inclusion of C1.) The logical device names L1 and P1 are also automatically generated, however the system manager may redefine them with the :LDEV command with any symbiont device type that he desires. The remaining 12 logical device names are selected by the system manager. It is best to select symbolic names that will be easy to remember.

It is advisable to define a logical device name for each symbiont device type, because all symbiont device types that are not given a logical device name are automatically defaulted to L1. In such cases, the batch, on-line, and ghost users must use the LDEV command or M:LDEV procedure to specify the symbiont device type for L1.

:OPLBLT This command is used to define the standard operational labels for the target system. Only one :OPLBLT command may be used although there may be numerous continuation records. The format of the command is as follows

```
:OPLBLT [(label,batch name,on-line name
     ,ghost name)][,(label,batch name,on-line
     name,ghost name)]...
```

where

label defines a monitor operational label comprising one or two alphanumeric characters. This label may not be the same as a device type (e.g., CR, CP) or a logical device name (e.g., C1). NO, MT, and SP are also invalid.

batch name specifies a physical device (e.g., LPA02), a device type (e.g., ME), or a logical device stream name (e.g., L1) to which the label is assigned for the batch mode. It may not be another operational label. If it is a physical device, it must not be a symbiont device.

on-line name specifies a physical device, a device type, or a logical device stream name to which the label is assigned for the on-line mode. (The restrictions for the batch name apply to the on-line name.)

ghost name specifies a physical device, a device type, or a logical device stream name to which the label is assigned for the ghost operations. (The restrictions for the batch name apply to the ghost name.)

For each label option, all three names must be specified (even if the label is one of the standard labels for which PASS2 has default assignments). In a batch only system, the on-line field must be included and should be set to NO.

The list of standard default operational labels and assignments is given in Table 68. (The labels appear in the order in which they appear in the tables generated.) Assignments are either directly to a device or to a logical device stream which in turn is connected to a physical device as determined by the :LDEV command.

Table 68 Standard Default Operational Labels and Assignments

Standard Operational Label	Batch	On-Line	Ghost
C	C1	ME	TY
OC	TY	ME	TY
LO	L1	ME	L1
LL	L1	ME	L1
DO	L1	ME	L1
PO	P1	NO	P1
BO	P1	NO	P1
LI	C1	NO	TY
SI	C1	ME	TY
BI	C1	NO	TY
SL	L1	ME	L1
SO	P1	NO	P1
CI	C1	NO	TY
CO	P1	NO	P1
AL	P1	NO	P1
EI	C1	ME	TY
EO	P1	NO	P1
UC	TY	ME	TY

:RES This command establishes definitions of resources which are used to control batch, on-line, ghost, and remote operations. Only one :RES command may be specified. The command has the form

:RES [(option)[,(option)]...]

The options are listed below. For those options in which a value must be specified, the meaning of "value" depends upon the particular resource. As an example, for tapes, it is "number of tape drives"; for disk packs, it is "number of private spindles"; for core, it is "number of K words".

RES,name specifies the name of a resource to which all subsequent options (until the next RES of end-of-command) apply. The name is either a name that was specified on a :DEVICE command (e.g., 9T) or a nondevice type (e.g., CO (core)). A symbiont device cannot be defined as a resource. A total of 15 RES options are permitted. If no RES option is specified, four entries are generated by default – CO, 9T, 7T, and SP. All options preceding the first RES option are ignored.

TOT,value specifies, in decimal, the total number of resources available (of the type being defined). TOT may be specified for device type resources, but must be specified for nondevice type resources.

BSUM,value specifies, in decimal, the sum of all resources (of the type being defined) that may be allocated to all concurrent batch jobs taken together.

OSUM,value specifies, in decimal, the sum of all resources (of the type being defined) that may be allocated to all concurrent on-line jobs taken together.

GSUM,value specifies, in decimal, the sum of all resources (of the type being defined) that may be allocated to all concurrent ghost jobs taken together.

BMAX,value specifies, in decimal, the maximum number of resources (of the type being defined) that may be allocated to an individual batch job.

OMAX,value specifies, in decimal, the maximum number of resources (of the type being defined) that may be allocated to an individual on-line job.

GMAX,value specifies, in decimal, the maximum number of resources (of the type being defined) that may be allocated to an individual ghost job.

BDEF,value specifies, in decimal, the default number of resources (of the type being defined) to be allocated to each batch job in those instances in which the user has made no specific request.

ODEF,value specifies, in decimal, the default number of resources (of the type being defined) which may be allocated to each on-line job in those instances in which the user has made no specific request.

GDEF,value specifies, in decimal, the default number of resources (of the type being defined) which may be allocated to each ghost job in those instances in which the user has made no specific request.

PASS2 verifies that the following relationships exist:

1. For device type resources, the number of devices defined via :DEVICE commands must be greater than or equal to the value specified for TOT. (If this is not the case, an error message is generated, TOT is set to the number defined via :DEVICE commands, and PASS2 continues.)

2. For each device type resource and each nondevice type resource,

   TOT ≥ BSUM ≥ BMAX ≥ BDEF

   TOT ≥ OSUM ≥ OMAX ≥ ODEF

   TOT ≥ GSUM ≥ GMAX ≥ GDEF

   (If this is not the case, an error message is generated, one of the values is reset as indicated in the error message description, and PASS2 continues.)

PASS2 always generates the following four entries with the default values as shown in Table 69 except when they are specified on the :RES command with other values specified.

Table 69. :RES Command Defaults

Option Name \ Resource Name	CO (core)	9T (tapes)	7T (tapes)	SP (private disk packs)
TOT	X'7FFF'	#†	#	#
BSUM	X'7FFF'	#	#	#
BMAX	X'10'	#	#	#
BDEF	X'C'	0	0	0
OSUM	X'7FFF'	#	#	#
OMAX	X'10'	#-1††	#-1	#-1
ODEF	X'C'	0	0	0
GSUM	X'7FFF'	#	#	#
GMAX	X'FF'	#-1	#-1	#-1
GDEF	X'FF'	#-1	#-1	#-1

†# is the total number of this type of device specified on :DEVICE commands.

††If #-1=0, the value is set to 1.

:BLIMIT / **:OLIMIT** / **:GLIMIT** These commands specify the system defaults and maximum values that are to be associated with each batch, on-line, or ghost job, respectively. The commands have the form

```
{:BLIMIT}
{:OLIMIT} [(option)][,(option)]...
{:GLIMIT}
```

where the options are

TIME,default,maximum specifies, in decimal number of minutes, the default limit and maximum limit for job execution time.

LO,default,maximum specifies, in decimal, the default limit and maximum limit on the number of pages of printer output from all shared processors involved in running a job.

PO,default,maximum specifies, in decimal, the default limit and maximum limit on the number of punched card records produced in running a job.

DO,default,maximum specifies, in decimal, the default limit and maximum limit on the number of pages of diagnostics produced in running a job (output through the M:DO DCB).

UO,default,maximum specifies, in decimal, the default limit and maximum limit on the number of pages of printed output from all the executing programs in a job.

TSTORE,default,maximum specifies, in decimal, the default limit and maximum limit for the number of granules of temporary RAD storage that may be used by a job.

PSTORE,default,maximum specifies, in decimal, the default limit and maximum limit for the number of granules of permanent RAD storage that may be used by a job.

TDISK,default,maximum specifies, in decimal, the default limit and maximum limit for the number of granules of temporary disk pack storage that may be used by a job.

PDISK,default,maximum specifies, in decimal, the default limit and maximum limit for the number of granules of permanent disk pack storage that may be used by a job.

FPOOL,default,maximum specifies, in decimal, the default limit and the maximum limit on the number of file blocking buffers to be allocated to a job. In general, each open disk file and each labeled tape DCB requires a blocking buffer. If an insufficient number of blocking buffers exists, they will be shared (at the price of reduced system performance). The default can be overridden by use of a POOL control command at run-time in the batch mode.

Note: At present, the default values for :GLIMIT options and the maximum values for :OLIMIT options are not used. To avoid later confusion it is recommended that identical values for default and maximum be used for all :GLIMIT and :OLIMIT options.

:ELIMIT This control command specifies the system unit increments for exit control processing. The command has the form

```
:ELIMIT (option)[,(option)]...
```

where the options are

TIME,value specifies, in decimal, the absolute unit for exit control execution time. The value is expressed in seconds and the default is 30.

LO,value specifies, in decimal, the increment for the number of processor pages output in running a job. The default is 20.

PO,value specifies, in decimal, the increment for the number of punched cards output in running a job. The default is 100.

DO,value specifies, in decimal, the increment for the number of diagnostic pages output in running a job. The default is 20.

UO,value specifies, in decimal, the increment for the number of user pages output in running a job. The default is 20.

TSTORE,value specifies, in decimal, the increment for the number of granules of temporary disk storage that may be used by a job. The default is 64.

PSTORE,value specifies, in decimal, the increment for the number of granules of permanent disk storage that may be used by a job. The default is 20.

:IMC This control command defines the installation management control (IMC) system limits. The labels in parentheses are those associated with the corresponding variables by the Control processor. The command has the form

```
:IMC (option)[,(option)]...
```

The options are (all values are in decimal):

MAXOL, value specifies the maximum number of on-line users allowed in the system. The default is 0. (OUM)

MAXB, value specifies the maximum number of concurrent batch users allowed in the system. The default is 1. (BUM)

MAXG, value specifies the maximum number of ghost jobs (exluding GHOST1) allowed in the system. MAXG must be in the range 255≥MAXG≥4. The default is 8. (GUM)

MAXG + MAXB + MAXOL = number of users allowed in the system and must not exceed 255. (UM)

BPRIO, value specifies the execution priority for all batch jobs. This priority is used for execution only and is not related to the scheduling priority. The value must be expressed in hexadecimal and must be in the range C0-FF. The default is FC. (SL:BPRIO)

OPRIO, value specifies the execution priority for all on-line jobs. This priority is used for execution only and is not related to the scheduling priority. The value must be expressed in hexadecimal and must be in the range C0-FF. The default is FC. (SL:OPRIO)

GPRIO, value specifies the execution priority for all ghost jobs. This priority is used for execution only and is not related to the scheduling priority. The value must be expressed in hexadecimal and must be in the range C0-FF. The default is FC. (SL:GPRIO)

BLOCK, value specifies the number of characters at which terminal output is blocked. The default is 40. (TB)

UNBLOCK, value specifies the number of characters at which terminal output is unblocked, where UNBLOCK≤BLOCK. The default is 8. (UB)

QUANTA, value specifies the time slice for compute-bound users. This value is in milliseconds. The default is 400. (QUAN)

MINTIME, value specifies the minimum time a user will be held in core before outswap. The decimal value is in milliseconds and must be less than or equal to QUANTA value. The default is QUANTA value. (SQUAN)

MINQUAN, value specifies a guaranteed time slice after a selection has been made and before a swap will occur. This value is in milliseconds. The default is 40. (QMIN)

COCBUF, value specifies the maximum number of COC buffers that can be allocated to a user in the type-ahead mode. COCBUF must be less than 256. A value of zero will yield unrestricted buffer allocation. The default is 15. (ONCB)

LOGTIME, value specifies (in minutes) the amount of time a user has to complete the log-on sequence. The default is 3. (OLTO)

INTIME, value specifies (in minutes) the amount of time a user has to input a line of data to COC when a read is pending. The default is 15. (OITO)

PI, value specifies the increment in units of 1/256 to be used in increasing the priority of jobs bypassed by the Multi-Batch Scheduler. The default is 0. (PI)

UCYL, value specifies the number of cylinders per user to be allocated as swap space on a disk pack. UCYL must be specified if, and only if, a disk pack is to be used as the swap storage device. The value of UCYL must be 1 or 2. The default is 1. If UCYL is specified for a RAD swapping system, a message will indicate that the specification is illegal and the specification will be ignored.

RASIZE, value specifies the number of table entries to be built for read-ahead facilities. One table entry is required for each concurrent read-ahead operation. The value is expressed in decimal and must be in the range 1-63. This option causes the read-ahead module (RA) to be included in the root of the monitor. (RA requires approximately 512 words.)

RAMAX, value specifies the maximum number of concurrent read-ahead operations to be allowed, where RAMAX ≤ RASIZE. The default is zero. (RAM)

RATO, value specifies the time, in milliseconds, after which an unused read-ahead block will be purged (time-out). The value must be expressed in decimal and must be in the range 0-32,767. The default is 10,000. (RATO)

File Maintenance Options

EXPIRE, days, hours specifies the system default used to establish expiration dates for files created by a user not specifying an explicit expiration date. To select a value of "never", this parameter must be omitted (the default is "never"). The maximum value that may be specified is 999 days, 23 hours and this will be used when the value specified exceeds this unit.

MAXEXPIRE,days,hours specifies the maximum retention period that may be specified by a user. To select a value of "never" this parameter must be omitted (the default is "never"). The maximum value that may be specified is 999 days, 23 hours and this will be used when the value specified exceeds this limit.

Note: The EXPIRE value specified must be equal to or less than the MAXEXPIRE value. If this is not the case the value for MAXEXPIRE will be set equal to the EXPIRE value.

BACKUPALL specifies that expired files will be backed up prior to deletion. If the parameter is omitted, files will not be backed up.

THRESHOLD,value specifies the critical point in initiating an automatic purge to prevent device saturation. The default is 500 granules.

:COC This control command specifies the characteristics associated with the COC devices. The COC command has the form

:COC (option)[,(option)] . . . , (COC)[,(option)] . . .

The first COC device is specified by the COC command word. Each additional COC device is specified by a COC keyword. All options preceding the first COC keyword apply to the command word (COC_0) while all options following the nth COC keyword apply to the nth COC (COC_n).

The options are

SS specifies that the Selectric® standard translation table is to be included in the monitor for a 2741 terminal.

SA specifies that the Selectric APL translation table is to be included in the monitor for a 2741 terminal.

ES specifies that the EBCD standard translation table is to be included in the monitor for a 2741 terminal.

EA specifies that the EBCD APL translation table is to be included in the monitor for a 2741 terminal.

DEVICE, ndd specifies, in hexadecimal, the number of a COC device.

OUT, value specifies the hexadecimal location of the COC output interrupt. The default is 61.

IN, value specifies the hexadecimal location of the COC input interrupt. The default is 60.

The SS, SA, ES, and EA options need be specified only once. The DEVICE option must be specified for every COC device. IN and OUT must be specified for the second and subsequent COC devices.

The input interrupts for all COC devices must be within one interrupt group.† The output interrupts for all COC devices must also be within one interrupt group. (The COC input and output interrupts may be in the same interrupt group.)

All COC input interrupts should be of higher priority than all COC output interrupts. COC0 input must have the highest priority of all COC interrupts. The location specified for OUT or IN determines the priority (the higher the location, the lower the priority). Generally, CP-V systems have only one COC device and use 60 and 61 as the input and output interrupt locations.

LINES, value specifies, in decimal, the number of lines. The default is 8.

BUFFERS, value specifies, in decimal, the number of 4-word buffers that will be pooled for COC I/O usage. The default is the highest line number. The recommended value is the number of users times 3.

RING, value specifies, in decimal, the COC input buffer size in words. The default is: (2 bytes per line for the first 30 plus 1 byte per line above 30) divided by 4. High speed terminals may saturate this buffer; therefore, when they are present on a system, a larger buffer size should be selected. The value must be less than or equal to 255.

7015,#,... specifies, in decimal, the line numbers that are attached to Xerox Model 7015 Keyboard printers. Each # field may be a single value or a range specification m–n where m<n. In the latter case, all numbers included in the range are used as line numbers.

2741,#,... specifies, in decimal, the line numbers that are attached to Model 2741 units. Each # field may be a single value or a range specification m–n where m<n. In the latter case, all numbers included in the range are used as line numbers.

HARDWIRE,#,... specifies, in decimal, the line numbers that are hardwired to terminals. Lines

®Registered trademark of the International Business Machine Corporation.

†External interrupts are described in the Sigma hardware reference manuals.

which are "hardwired" will not be timed out (except 2741s logging on), nor will they log on automatically. All lines must be individually specified but need not be in ascending order. Each # field may be a single value or a range specification m–n where m<n. In the latter case, all numbers included in the range are used as line numbers.

TYPE,value,#,#... where value specifies, in decimal, the timing algorithm number that the line number(s) following the value specification use. The value may be between 0 and 7. The six timing algorithms that are supplied with CP-V are listed in Table 70. (The system manager may create additional algorithms.) The default TYPE value for lines that are listed on the 2741 option is 1. The default TYPE value for all other lines is 0. All line numbers must be individually specified, but need not be in ascending order. Each # field may be a single value or a range specification m–n where m<n. In the latter case, all numbers included in the range are used as line numbers.

RATE,value,#,#... where value specifies, in decimal, the speed of the terminals (in characters per second) associated with the line number(s) following the value specification. The value may be in the range 1 to 255. One RATE option is used for each terminal speed on the COC. The default RATE for lines that are listed on the 2741 option is 15. The default RATE for all others is 10. All line numbers must be individually specified, but need not be in ascending order. Each # field may be a single value or a range specification m–n where m<n. In the latter case, all numbers included in the range are used as line numbers.

Model 33 is the default terminal.

:MON This command defines various monitor and CPU parameters for the target system and has the form

:MON (option)[,(option)]...

where the options are

AVGSER,n specifies, in decimal, four times the average number of private disk packs required per batch job. The default is 1, indicating an average one-fourth disk pack per job.

[BIG9 / SIG9] specifies that the target system is a Sigma 9 computer system and that the module name S9TRAPS is to be added to the SPEC:HAND file. In addition, the data and variables required for S9TRAPS are included in M:CPU. SIG9 specifies a target system whose memory is 128K or less BIG9 specifies that the target system is a large memory map (greater than 128K) Sigma 9 and causes SYSGEN to generate internal tables that accommodate physical memory references greater than 128K. SIG9 may be specified without BIG9 being specified. However, if BIG9 is specified without SIG9 being specified, SIG9 will automatically be assumed and a message noting this will be generated. A system generated with a SIG9 option will operate on a Sigma 6, 7, or 9; however, a system generated with a BIG9 option will operate only on a Sigma 9. The default is a Sigma 6 or 7 and the system generated will only operate on the Sigma 6 or 7.

INFILE,n specifies, in decimal, the number of input symbiont files that may be queued waiting for entry into the system. The default is 20.

OUTFILE,n specifies, in decimal, the number of output symbiont files that may be queued waiting for devices. The default is 32.

CORE,size specifies, in decimal units of K (where K = 1024), the size of core storage in the target system. The default is 128K, which is the recommended minimum in CP-V since it dynamically adjusts itself to the actual size of machine at boot time.

QUEUE,size specifies, in decimal, the maximum number of I/O operations that may be queued at one time. The recommended value is a total of 1 per nondisk type device and 2 per disk type device. Although a smaller value will work, the minimum should be a total of 1 per channel and 1 per disk type device. For standard systems, the value should be 25–30. In addition, the MXSTRM value (see :SDEVICE command) should be added to this value for remote processing systems. The default and minimum are 10.

MPOOL,size specifies, in decimal, the number of 34-word buffers to be pooled for use by the monitor. It is recommended that the number of MPOOLs be one-half to three-fourths the number of QUEUE entries. For standard systems, the value is between 14 and 22. One extra MPOOL should be defined for each RB device included in the system. The default is 5.

CPOOL,size specifies, in decimal, the number of 40-word buffers to be pooled for symbiont context block use. The default is 3. The recommended value is the number of symbiont I/O devices. For remote processing systems, the value should be increased by one for each 7670 RB line and by a number that is approximately three-fourths the maximum number of peripheral devices that could exist at concurrently logged on IRBTs.

Table 70. Timing Algorithms

<table>
<tr><th>Timing Algorithm Number</th><th>Usage</th><th>Idles[†]</th></tr>
<tr><td>0</td><td>Teletype Models 33, 35, and 37 and alphanumeric displays.</td><td>None</td></tr>
<tr><td>1</td><td>IBM 2741 and 2741-compatible equipment.</td><td>Before carriage return = name
After carriage return = (curpos[††]+15)/10
After tab character = (new position-old position+15)/10</td></tr>
<tr><td>2</td><td>Execuport and Dataport terminals.</td><td><table>
<tr><th></th><th>0-10 cps[†††]</th><th>11-15 cps</th><th>16-30 cps</th><th>31-60 cps</th><th>61- cps</th></tr>
<tr><td>Before carriage return</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>After carriage return</td><td>1</td><td>4</td><td>8</td><td>12</td><td>16</td></tr>
<tr><td>After tab character</td><td>1</td><td>1</td><td>2</td><td>4</td><td>8</td></tr>
</table></td></tr>
<tr><td>3</td><td>Memorex terminals</td><td><table>
<tr><th></th><th>0-10 cps[†††]</th><th>11-15 cps</th><th>16-30 cps</th><th>31-60 cps</th><th>61- cps</th></tr>
<tr><td>Before carriage return</td><td>7-curpos[††]</td><td>10-curpos</td><td>20-curpos</td><td>40-curpos</td><td>40-curpos</td></tr>
<tr><td>After carriage return</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>After tab character</td><td>1</td><td>1</td><td>2</td><td>4</td><td>8</td></tr>
</table></td></tr>
<tr><td>4</td><td>This algorithm is a combination of the others and may be used to ensure that an inexperienced user can utilize the system without any character loss. It also supports an experienced user until a change in terminal type can be entered. It is suggested that installations with mixed types of high-speed terminals use this algorithm as the default for high-speed lines.</td><td><table>
<tr><th></th><th>0-10 cps[†††]</th><th>11-15 cps</th><th>16-30 cps</th><th>31-60 cps</th><th>61- cps</th></tr>
<tr><td>Before carriage return</td><td>7-curpos[††]</td><td>10-curpos</td><td>20-curpos</td><td>40-curpos</td><td>40-curpos</td></tr>
<tr><td>After carriage return</td><td>1</td><td>4</td><td>8</td><td>12</td><td>16</td></tr>
<tr><td>After tab character</td><td>1</td><td>1</td><td>2</td><td>4</td><td>8</td></tr>
</table></td></tr>
<tr><td>5</td><td>This algorithm is used for terminals that require a number of idles roughly proportional to the carriage movement distance. It may be used for Teletypes and other equipment of similar mechanical design, and is sometimes a better algorithm than number 0 for such equipment.</td><td><table>
<tr><th></th><th>0-10 cps[†††]</th><th>11-15 cps</th><th>16-30 cps</th><th>31-60 cps</th><th>61- cps</th></tr>
<tr><td>X</td><td>60</td><td>50</td><td>18</td><td>15</td><td>15</td></tr>
</table>Before carriage return = none
After carriage return = (curpos+15)/X
After tab character = (new position-old position+15)/X</td></tr>
</table>

[†]Many high-speed terminals require a delay before sending a carriage return, after sending a carriage return, or after sending a tab character. In such a case, the COC handler must send "idle" characters, the number of which depends upon line speed, carriage position, and characteristics of the particular terminal.

[††]Current carriage position.

[†††]Characters per second.

CFU, size specifies, in decimal, the number of 19-word buffers to be pooled for current file users. The default is 40. The minimum is 8. Starting with the B00 release of CP-V, a more economical use is made of the CFU area by the file management system. The estimated effect of this change is to provide approximately 60 percent more effective use of the CFU area. Thus, the recommended value is 1.2 times the number of users (as defined by the :IMC control command). It is recommended that users of previous releases of CP-V scale this option down to 60 percent of the value previously used if the number of users to be supported remains constant.

ORG, value specifies, in hexadecimal, the load origin of the monitor. This value should be the address of the first unused even-numbered interrupt, since the monitor must be biased above the highest interrupt. If there are no special interrupts, the origin should be the first even-numbered address above the highest COC interrupt. The default is 62.

MPATCH, size specifies, in decimal, the number of word locations to be reserved for modification of the monitor (i.e., a patch area). The default is 0. The required minimum is 512.

SITE, site-id specifies a one to eight character site identification. The site-id becomes a part of the title printed on each page of an ANLZ dump. If less than eight characters are specified, the name will be left justified and blank filled. If more than eight are specified, only the first eight will be used.

[ANSDET | ANSPROT] specifies that tape handling is to detect possible overwrites of unexpired ANS tapes and missing write rings in output tapes. If ANSPROT is specified, tape handling will operate in the "protective mode" (see Glossary). If ANSDET is specified, the "semi-protective mode" (see Glossary) is used. If neither is specified, the system handles ANS tapes but no special checks are made. If either is specified, tapes mounted as IN or INOUT are given special handling. Tapes mounted as IN are flagged as such and may not be written on unless they are at the load point (in which case they are processed as though they are just going through the MOUNT process). Tapes mounted as INOUT must have a write ring; otherwise, a message will be sent to the operator's console. The operator can override the condition if the system is in the semi-protective mode.

ENQ, size specifies that the enqueue/dequeue facility is to be part of the target system. Size specifies, in decimal, the number of doublewords to be allocated for the ENQ tables. Size should be roughly 18-24 times the number of projected concurrent enqueue users. If lengthy qnames and snames are to be used, size should be larger.

:SPROCS This control command defines the characteristics of shared processors or monitor overlays and optionally reserves space for the dynamic addition or replacement of both. The :SPROCS command has the form

:SPROCS [(name[, option]...)][, (name[, option]...)]...

where name may be

MON followed by the names of overlays to be added to the automatic default overlays which are KEYIN, LTAPE, DEBUG, OPEN, CLOSE, STEPOVR, LDLNK, MUL, MISOV, RMAOV, and IODTYPR. The names must be less than or equal to seven characters in length. Only those portions of the monitor that have been specifically coded as overlays must be specified. This is necessary because a mapped overlay scheme is used.

processor name followed by flags and/or a number in the form

(procname, flags, number)

or

(procname, flags)

or

(procname, number)

The number specifies, in decimal, the number of overlays that will be associated and the following flags are possible.

S specifies that this processor resides in the special shared processor's virtual memory.

J specifies that special JIT (job information table) access is to be given.

D specifies that the processor is to be treated as a debugger. If specified, S is also assumed.

P specifies that the processor is to be treated as a public (core) library, and if specified, S is assumed. The name must be the form :Pi ($0 \le i \le 9$). If valid, the name :Pi will be changed to :Pii.

M specifies that maximum memory for this processor, i.e., all of the available core is given to this processor when in execution.

T specifies that this is a command processor and that it is accessible by the terminal users.

B specifies that this is a command processor and that it is accessible by batch users.

G specifies that this is a command processor and that it is accessible by ghost users.

C specifies that this is a command processor and that it is accessible by terminal, batch, and ghost users.

The flags S, J, D, P, M, T, B, G, and C may also be grouped syntactically without intervening commas (e.g., SJ, etc., up to a maximum of four characters). If D or P is specified, S is redundant.

Table 71 is the default list of standard processors. To change the characteristics of one of the standard processors, one of the following should be specified:

(REPLACE, procname, flags, number)

or

(REPLACE, procname, flags)

or

(REPLACE, procname, number)

Procname, flags, and number are described above. The processors in Table 71 can be specified on the :SPROCS command only in conjunction with REPLACE.

For those installations in which processors and/or monitor overlays are to be dynamically added or replaced, the following name and options will be required as applicable:

MOSPACE, number specifies, in decimal, the number of spare slots to be reserved in the monitor overlay area for additions or replacements. The default is 0 and the range is unlimited.

POSPACE, number specifies, in decimal, the number of spare slots to be reserved in the processor overlay area for additions or replacements. The default is 0 and the range is unlimited.

PSPACE, number, size specifies, in decimal, the number of spare slots to be reserved in the processor area and the total number of pages required for one spare processor. This includes all data, procedures, and overlays. The default size is 6 pages. The permissible range of size is 6 to 256 pages. The recommended size is 64 pages. If PSA has been specified on a :DEVICE command defining a disk pack, the size parameter must be less than or equal to the number of granules per physical cylinder (i.e., 60 for the 7242/7270 or 110 for the 7260/7275). Otherwise, it is set to the appropriate value and a message is produced. Beware that PSA space is consumed by these spare slots.

Example:

```
:SPROCS (MON,BLOB),(GLOP,M,3),(POSPACE,4),;
:          (PSPACE,4,64),(MOSPACE,3)
```

Table 71. Standard List of Shared Processors

Processor Name	J S D P M T G B C	No. of Overlays
TEL	1 1 0 0 1 1 0 0 0	0
CCI	1 0 0 0 1 0 0 1 0	0
GHOST1	1 0 0 0 1 0 1 0 0	0
LOGON	1 0 0 0 1 1 1 1 0	0
LINK	0 1 0 0 1 0 0 0 0	0
DELTA	0 1 1 0 0 0 0 0 0	0
:P00	0 1 0 1 0 0 0 0 0	0
:P11	0 1 0 1 0 0 0 0 0	0
EDIT	0 0 0 0 0 0 0 0 0	0
PCL	0 0 0 0 0 0 0 0 0	0
BASIC	0 0 0 0 0 0 0 0 0	0
METASYM	0 0 0 0 0 0 0 0 0	2
LOADER	0 0 0 0 1 0 0 0 0	7
FORT	0 0 0 0 0 0 0 0 0	3
FILL	1 0 0 0 1 0 0 0 0	0
RUNNER	1 1 0 0 0 0 0 0 0	0
BATCH	0 0 0 0 0 0 0 0 0	0
DRSP	0 0 0 0 1 0 0 0 0	0
LDEV	1 0 0 0 0 0 0 0 0	0

The monitor overlay BLOB will be added to the standard monitor list. The shared processor, GLOP, with three overlays and obtaining all available core while in execution, will be added to the standard shared processor list. Additionally, three slots have been reserved for adding monitor overlays dynamically. Four slots have been reserved for shared processors each of which will have 64 pages reserved for them.

:PART This control command is used to define the number of partitions which may be used and their resource values. The :PART command has the form

:PART [(option)][, (option)]. . .

The options are listed below. All values must be expressed in decimal. For all options which specify minimum and maximum values, the minimum value must be less than or equal to the maximum value.

TIME, min, max specifies, in minutes, the time range of the partition. The default is 0, 5.

QUAN, value specifies, in milliseconds, the quantum time of the partition. The default is 400.

name, min, max specifies the name of a resource (defined via the :RES command) and limits on that resource which are to be used for partition control. The range of permissible values is 0 to the maximum number available for batch allocation (i.e., the value specified for BMAX for the resource on the :RES command). Resource names are always generated for CO (core), 9T, 7T, and SP. The minimum and maximum values default to zero. A total of eleven more resource names may be specified to be used in batch job scheduling.

{HOLD | SWAP} specifies whether the partition is to be held in core or is available for swapping. The default is SWAP. If both keywords are present, then SWAP will be used.

{LOCK | UNLOCK} specifies whether the partition is to be available for job selection or is locked from further use. The default is UNLOCK. If both keywords are present, then UNLOCK will be used.

PART, value specifies the number of a partition to which all subsequent options until the next PART or end of record will apply. All options preceding the first PART are default values to be used when explicit specifications are not given. An option not specified either preceding or following a partition number will be set to the standard SYSGEN default. The total number of partitions defined is equal to the highest PART number specified and must be less than or equal to 16. If no PART value is specified the total number of partitions is 3 by default.

Example:

```
:PART  (LOCK),(CO,0,10),(TIME,0,5),;
:      (PART,5),(PART,4),(TIME,6,8),;
:      (UNLOCK)
```

All options except LOCK, CO, TIME for all partitions are set to their standard SYSGEN limits. The time range will be set to 0,5 for all partitions except partition 4 which will have the time range 6, 8. Also partition 4 is to be the only partition available for job selection. There will be a total of five partitions defined.

:FRGD This control command specifies that the system is to have real-time facilities. The format of the command is

:FRGD [(option)][, (option)]

where the options are:

RESDF, size specifies the default size in pages of the dedicated foreground memory area to be allocated at system initialization. This value may be overridden by the operator at system initialization.

NINT, value specifies, in decimal, the maximum number of interrupts that can be centrally connected at one time. This value also specifies the number of Interrupt Control Blocks (ICBs) to be allocated. The ORG option on the :MON command should be used to bias the monitor above any desired hardware interrupt levels.

:INTLB This command associates a 2-character label with an interrupt address. The format of the command is

:INTLB (label, loc)[, (label, loc)]. . .

where

label specifies a 2-character label.

loc specifies the absolute hexadecimal interrupt location or a pseudo-interrupt number to be associated with the label. The permissible values for an interrupt address are in the hexadecimal range 58-59 or 5F-140. The permissible values for a pseudo-interrupt number are in the hexadecimal range 1000-7FFF.

:FAUTH The feature authorization control command specifies that users' access to certain system features is to be controlled by the Super and Control processors. If a special feature (defined below) is not specified on this command, all users are to have access to that feature.

Note that this command does not regulate the inclusion of special features in the system. For example, the special feature Enqueue/Dequeue is included in the system by specifying the ENQ option on the :MON command.

The format of the command is:

:FAUTH (feature)[, (feature). . .]

where feature specifies a special feature. The current special features are:

EQ – enqueue/dequeue facilities

JE – M:JOB job entry

RP – remote processing

PASS2 CONTROL COMMAND SEQUENCE

PASS2 reads all of the input commands and creates an intermediate temporary keyed file. The records (the : commands) of the file are then accessed in the order that PASS2 requires to build the dynamic monitor tables. The only restriction placed upon the order of the commands to PASS2 concerns the :CHAN and :DEVICE commands. At least one :CHAN must be used and each :CHAN must precede the :DEVICE command (or commands) to which it applies.

SYSGEN PASS2 OVERLAY STRUCTURE

TREE STRUCTURE OF PASS2

Figure 92 is the functional tree structure of PASS2 (the names identify the necessary element files).

LOAD MODULES AND FILES GENERATED BY PASS2

Table 72 identifies the load modules, data files, or element files[†] that are generated by the corresponding PASS2 control commands.

PASS2 MESSAGES

All PASS2 messages (Table 73) are output on the LL device. When PASS2 attempts to continue (unless otherwise specified), it will search for a closing parenthesis ")" and continue processing from that point on. When an error message implies an error within a processor, this could also mean that there is not enough core to generate the current load module (LM).

[†]The term element file refers to a ROM (relocatable object module).

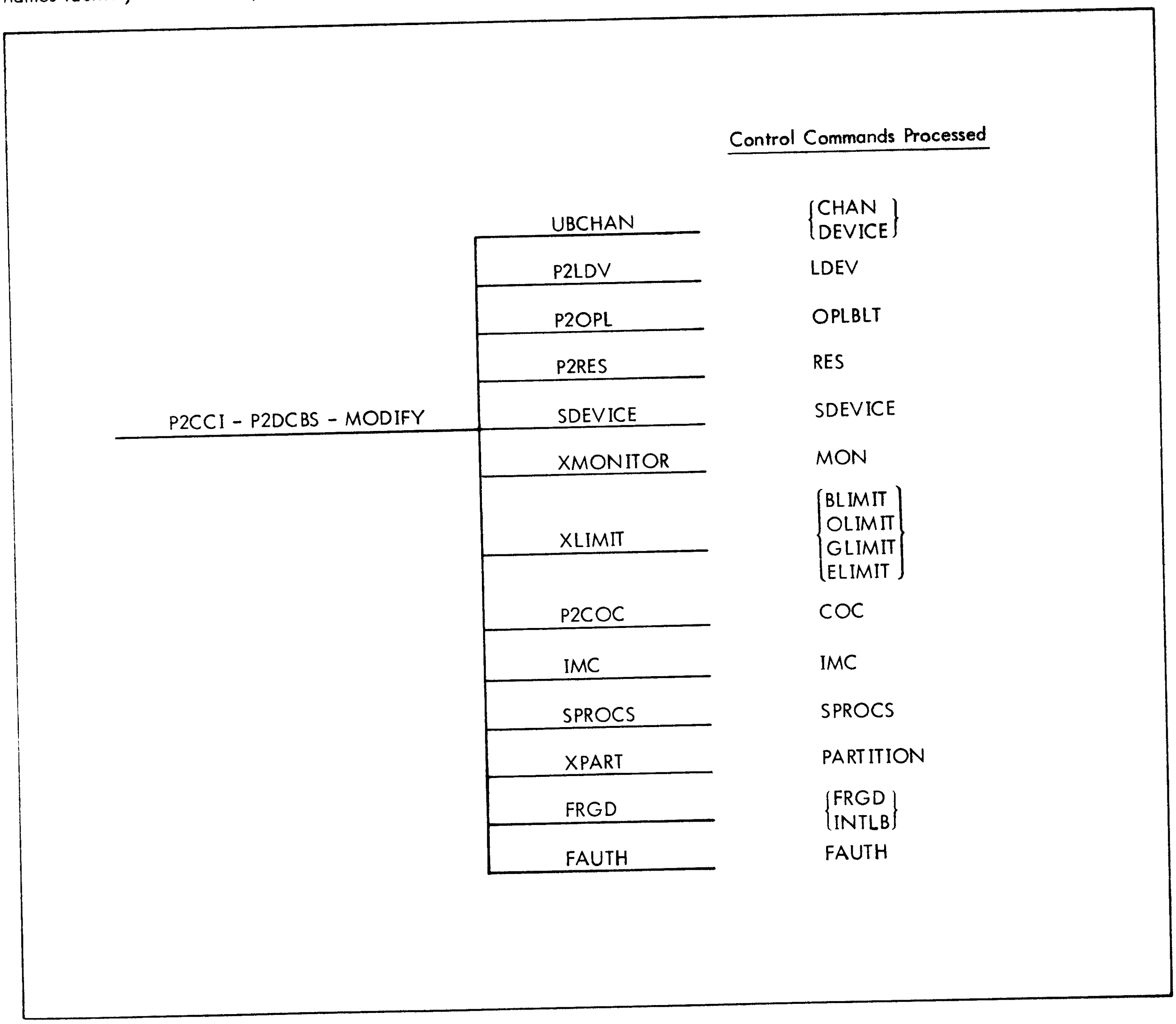


Figure 92. Tree Structure of PASS2

Table 72. PASS2 Load Modules and Files

PASS2 Commands	Load Module/Element File Name
CHAN DEVICE	IOTABLE (load module) SG:FLG (load module) SG:OPNM† (load module) SG:OPX† (load module) SPEC:HAND† (data file) M:DCTMOD (load module) M:HGP (load module)
SDEVICE	M:SDEV (load module) SG:PNT† (load module)
RES	SG:RNT (load module) SG:RCT (load module) SG:RTY (load module) SG:OPNM† (load module) SG:OPX† (load module)
LDEV	SG:OPNM† (load module) SG:OPX† (load module)
OPLBLT	SG:OPNM† (load module) SG:OPX† (load module) SG:FLG (load module)
MON	M:CPU (load module) M:SYMB (load module) M:BIG9 (load module) MON::ORG (element file)
BLIMIT GLIMIT OLIMIT	SG:LNT (load module) SG:DLNT (load module)
ELIMIT	M:ELIMIT (load module)
COC	M:COC (load module)
IMC	M:IMC (load module)
SPROCS	M:SPROCS (load module)
PART	M:PART (load module)
FRGD	SPEC:HAND† (data file) M:FRGD (load module)

†These modules are built from information from several commands.

Table 73. PASS2 Messages

Message	Description
General Messages	
***CANNOT READ CONTINUATION RECORD–PASS2 ABORTED	The card image following a continuation character is incorrect or out of place.
***CC IGNORED, PREVIOUS CC OF THIS TYPE ACCEPTED	The current control command type has already been encountered and processed. Only one set of a specific type of command is allowed in a run of PASS2. PASS2 continues to the next control command.
***DIFFICULTY PROCESSING PASS2 COMMANDS	In attempting to build the intermediate keyed file of : commands, PASS2 has encountered an abnormal/error condition. The I/O error code is output and PASS2 exits to the monitor.
***DUPLICATE CC OTHER THAN DEVICE/CHAN	A duplicate control command other than a :DEVICE or :CHAN command was encountered. PASS2 aborts.
.....END OF PASS2.....	The end of PASS2 has been reached. PASS2 exits to the monitor.
***ERROR IN M:EI FILE	The M:MODNUM file cannot be opened.
***ERROR PROCESSING COMMANDS FROM KEYED FILE	In attempting to access the records from the intermediate keyed file of : commands, PASS2 has encountered an abnormal/error condition. The I/O error code is output and PASS2 exits to the monitor.

Table 73. PASS2 Messages (cont.)

Message	Description
***I/O ERR/ABN = xx/xx	An I/O error occurred in attempting to build or access records from the intermediate keyed file of : commands. The error code is specified by xx/xx.
***NO COC COMMAND – A BATCH ONLY SYSTEM BEING GENERATED	Self-explanatory. PASS2 continues.
.....PASS2 CCI IN CONTROL.....	PASS2 has been entered.
***PASS2 DELAYED ABORT DUE TO FATAL ERROR	An error has occurred in processing PASS2 commands that has been determined to be fatal but from which PASS2 has attempted to continue processing commands to provide syntax analysis of nonaffected commands. An example of such an error is the omission of a handler for a nonstandard device. Some errors may have been produced as a result of the delayed abort condition and correction of the initial problem may correct subsequent problems. PASS2 issues an error exit.
***RECORD EXCEEDS 80 CHARACTERS	A card image of more than 80 characters was processed. PASS2 aborts.
***UNKNOWN OR MISPLACED CC	The current control command is unknown or out of order. PASS2 continues to next control command.
***UNKNOWN TYPE CP-V USED	The type of PASS2 is unknown. PASS2 defaults to CP-V.
CHAN/DEVICE Messages $	 When $ appears without additional messages, it indicates that there is a syntax error. PASS2 tries to continue.
***CANNOT BUILD SGP OR NON-RESIDENT HGP	The available core is not sufficient for generating these tables.
***DEVICE TYPE yy ILLEGAL	A DEVICE control command yyndd field contained a "NO", "MT", "SP" as its yy. PASS2 tries to continue.
***HGP CANNOT BE FORMED FOR yyndd	A DEVICE yyndd command (where yyndd is for a DC or DP device) contained a syntax error for which no defaults can be taken. PASS2 tries to continue.
***HGP TABLE FULL	The core area allocated for HGP tables for RAD or disk pack devices was not large enough. PASS2 aborts.
***INSUFFICIENT PAGES AVAILABLE	The available core was not large enough for the allocation required by generation of the load module. PASS2 aborts.
***LOAD MODULE CANNOT BE GENERATED	This message is produced in conjunction with the messages INSUFFICIENT PAGES AVAILABLE and ONLY xxxx PAGES OBTAINED. PASS2 aborts.
***MREC/MXREC VALUE INVALID – SYSGEN DEFAULT USED	The value for either MREC or MXREC exceeds 255. PASS2 substitutes 1 for MREC and 255 for MXREC and continues.
***'NAME' OR SYNTAX INVALID	A CHAN control command option field has a syntax error, or the DEVICE control command contained a syntax error or invalid name for the handler option. PASS2 tries to continue.

Table 73. PASS2 Messages (cont.)

Message	Description
***NCYL MISSING FOR MOVEABLE HEAD DISK DEVICE	In defining a nonstandard disk device, the parameter NCYL was omitted. PASS2 aborts.
***NGC>255 – 55 USED FOR 7261	The CYLIN value is too large. PASS2 continues.
***NGC>255 – 30 USED FOR 7242/7271	The CYLIN value is too large. PASS2 continues.
***NO CHAN/DEVICE INFO	No CHAN and DEVICE control commands have been encountered. PASS2 tries to continue to the next command that is not a CHAN or DEVICE command.
***NO DEVICE FOR CHAN	A CHAN control command has been encountered without any device definitions for the channel. PASS2 tries to continue.
***NO DISC DEFINED	No RAD or disk pack was defined by a DEVICE control command.
***NO HANDLER – DEVICE IGNORED – DELAYED ABORT CONDITION	The HANDLER option was not present for a nonstandard device. Although this is an abort condition, PASS2 will attempt to continue processing subsequent commands for syntax analysis but will issue an error exit at the end. Note that subsequent errors may be a function of this error condition.
***NO PER DEFINED	Self-explanatory. PASS2 continues.
***NO PSA DEFINED	Self-explanatory. PASS2 continues.
***NTPC MISSING FOR MOVEABLE HEAD DISK DEVICE	In defining a nonstandard disk device, the parameter NTPC was omitted. PASS2 aborts.
***ONLY xxxx PAGES OBTAINED	This message follows the messages HGP TABLE FULL and INSUFFICIENT PAGES AVAILABLE. xxxx is the number of pages that were available to build the load module. PASS2 aborts.
***ONLY PFA VALID ON PRIVATE OR CYLIN ALLOCATED DEVICE	PER and PSA are not allowed. PFA is set to the pack size.
***PER STARTING SECTOR EXCEEDS 16 BITS – PER SET TO 0	The starting sector address of PER exceeds the halfword field in the HGP and therefore is set to 0. (Actually, the PSA field overflowed and caused the PER field to overflow.) This condition can only occur when a 7265 or 7275 disk pack is specified.
***PFA STARTING SECTOR EXCEEDS 16 BITS – PFA SET TO 0	The starting sector address of PFA exceeds the halfword field in the HGP and therefore is set to 0. (Actually, the PER field overflowed and caused the PFA field to overflow. If this message is output in conjunction with the message 'PER STARTING SECTOR EXCEEDS 16 BITS – PER SET TO 0', then the PSA field overflowed and this caused both the PER and PFA fields to overflow.) This condition can only occur when a 7265 or 7275 disk pack is specified.
***PRIVATE DISK HGP EXCEEDS ONE PAGE	The value used for the number of granules per logical cylinder has caused the resulting HGP to exceed one page in length. PASS2 aborts.
***PSA DEFINED ON RAD, NOT ALLOWED ON DP	A disk pack cannot be used for swap storage if a RAD is also being used for swap storage. PASS2 tries to continue.

Table 73. PASS2 Messages (cont.)

Message	Description
***PSA+PER+PFA>SIZE	The actual size is used and truncation occurs in the area exceeding the size limit.
***PSA VALUE TOO LARGE – MAX VALUE USED PFA INCREMENTED	In defining the PSA area on a disk pack, the number of tracks specified exceeded X'3FC' for a 7242 or X'21C' for 7260/7265. PSA is set to the appropriate maximum value and the extra tracks are allocated as PFA. PASS2 continues.
***PASS2 UNABLE TO CONTINUE	Self-explanatory. PASS2 aborts.
***TROUBLE WITH M:MODNUM FILE OR MODNUM COMMAND. PASS2 UNABLE TO CONTINUE	An I/O error occurred.
***VALID 'CHAN' CC MUST PRECEDE 'DEVICE' CC	Self-explanatory. PASS2 continues.
***PSA INCREMENTED FOR DP SWAPPER – PFA/PER DECREMENTED	The number of tracks of PSA for a disk pack swapper was not evenly divisible by the number of tracks per physical cylinder. PSA was incremented so that the value would be evenly devisible and PFA and PER were correspondingly decremented. PASS2 tries to continue.
***PSA MUST BE 7212/7232 DISC PACK – PSA IGNORED	PSA has been specified on a device that is not recognized. PASS2 continues.
***PSA PREVIOUSLY DEFINED ON DISK PACK – PSA IGNORED	The PSA option may only be specified for one disk pack device. PASS2 continues.
***PSA/PER INVALID ON CYLIN ALLOCATED DEVICE – PSA/PER IGNORED	Self-explanatory. PASS2 continues.
***SIZE MISSING FOR FIXED HEAD DISK DEVICE	In defining a nonstandard disk device, the SIZE parameter was omitted. PASS2 aborts.
***SPEC:HAND CANNOT BE GENERATED	An abnormal condition prevents the creation of the SPEC:HAND file. PASS2 aborts.
***SS/NSPT/SIZE MUST BE NON-ZERO	Either a new disk device has not been fully defined or some model or controller number is incorrect. Therefore the device is new to PASS2 and the defaults cannot be used. PASS2 aborts.
***SUM OF PSA+PER+PFA>SIZE	There was a conflict in the summation of the given list of variables and the defined RAD or disk pack size. The message may appear several times for a given RAD (i.e., if the conflict is determined after the summation of PSA+PER, then the message will appear for this summation and once for each of the remaining summations). The processor continues.
***SYNTAX ERROR	This message appears in conjunction with the $ message. PASS2 tries to continue.
***SYNTAX ERROR DUAL OPTIONS USED	The closing double parenthesis on the DUAL option of the CHAN command are in error. However, the preceding option has been correctly processed and will be used. UBCHAN continues.

Table 73. PASS2 Messages (cont.)

Message	Description
***THIS DISC ALREADY DEFINED	A DEVICE control command defined a RAD or disk pack device (i.e., yyndd) that had already been defined. PASS2 tries to continue.
***UNKNOWN DEVICE yyndd FOR CH	The channel designated is out of range.
***WARNING DEVICE SIZE NOT EVENLY DIVISIBLE BY NGC – REMAINDER IGNORED	The size of the device computed to total number of granules is not evenly divisible by the value representing the number of granules per logical cylinder (NGC). PASS2 continues.
***WARNING INCORRECT MOD. # OR CONT. #. ZERO HAS BEEN SUBSTITUTED FOR BOTH	PASS2 continues. However, the system will have to be patched with the correct model and controller numbers.
***WARNING NEW DISK PACK USED AS SWAPPER	A nonstandard disk pack has been used as the swapper.
***VALID 'CHAN' CC MUST PRECEDE 'DEVICE' CC	A DEVICE control command not preceded by a CHAN control command was encountered. PASS2 tries to continue.
SDEVICE Messages	
***MXSTRM TOO LARGE – SET TO 128	The value specified for MXSTRM exceeds 128. MXSTRM is set to 128 and PASS2 continues.
***MXSTRM TOO SMALL – SET TO 3*#RBTS	In a system with remote devices defined, either MXSTRM was not specified or the value specified for it was too small. MXSTRM is set to three times the number of RBTs defined and PASS2 continues.
***NO SYMBIONT DEVICES DEFINED	There is missing or incorrect information on the :SDEVICE command.
**xxxx UNKNOWN – OPTION IGNORED	The character string xxxxx does not identify a device that was defined via the :DEVICE command. The option is ignored and PASS2 continues.
RES Messages	
***xxxx A SYMBIONT DEVICE RESOURCE IGNORED	Symbiont devices cannot be defined as resources. PASS2 continues.
***xxxx yyyyyyyy DEFAULT>MAX OPTION DEFAULTED	For the resource xxxx, the default is greater than the maximum for the mode specified by yyyyyyyy (batch, on-line, or ghost). The default is set as follows: If the resource is CO, it is set as indicated in Table 69. For all other resources, the default is set to zero for the batch or on-line mode or is set to the value specified for the maximum for the ghost mode. PASS2 continues.

Table 73. PASS2 Messages (cont.)

Message	Description
***xxxx yyyyyyyy MAX>SUM OPTION DEFAULTED	For the resource xxxx, the maximum value is greater than the sum for the mode specified by yyyyyyyy (batch, on-line, or ghost). The maximum is set as follows: If the resource is CO, it is set as indicated in Table 69. For all other resources, the maximum is set to the value specified for sum for the batch mode or to sum-1 for the ghost or on-line mode. PASS2 continues.
***NO INVALID RESOURCE – RESOURCE IGNORED	NO may not be specified as a resource. PASS2 continues.
***xxxx RESOURCE ALREADY DEFINED – OPTION IGNORED	The resource specified by xxxx was specified more than once on the :RES command. The first specification is used. PASS2 continues.
***xxxx yyyyyyyy SUM>TOT OPTION DEFAULTED	For the resource xxxx, the value specified for sum is greater than TOT for the mode specified by yyyyyyyy (batch, on-line, or ghost). The value for sum is set as follows: If the resource is CO, it is set as indicated in Table 69. For all other resources, sum is set to TOT for all modes. PASS2 continues.
***TOO MANY RESOURCES ONLY FIRST 15 USED	Only 15 resources can be specified. The first 15 are used and the rest are ignored. PASS2 continues.
***xxxx # TOTAL > # DEFINED ON SYSTEM – # DEFINED USED	For the resource xxxx, TOT is greater than the number defined in the :DEVICE command. (This message only applies to device type resources.) The number defined on :DEVICE is used for TOT. PASS2 continues.
***xxxx TOTAL MISSING, RESOURCE IGNORED	The TOT option must be specified for a nondevice type resource. PASS2 continues.
LDEV Messages	
***> 15 LDEV ENTRIES – ADDITIONAL ONES IGNORED	A maximum of 15 logical device names may be defined on the :LDEV command. The first 15 definitions are used and PASS2 continues.
***C1 CANNOT BE ASSIGNED – ENTRY IGNORED	The logical device name C1 is always automatically generated by the system and should not be specified on the :LDEV command. PASS2 continues.
***xx DEVICE TYPE NOT SYMBIONT – ENTRY IGNORED	The device type xx was not specified as a symbiont device on a :SDEVICE command. PASS2 continues.
***NAME A TYPMNEMONIC – NAME IGNORED	The name cannot be a type mnemonic. PASS2 continues.
***NAME DUPLICATED – FIRST ONE IGNORED	A logical device name was specified twice, PASS2 continues.

Table 73. PASS2 Messages (cont.)

Message	Description
***xx SYMBIONT DEVICE TYPE NOT SPECIFIED – DEFAULTED TO L1	A logical device name was not defined for the symbiont device type xx. It is given the logical device name L1 by default. PASS2 continues.
OPLBLT Messages	
***xxxxx ASSIGNMENT TO SYMBIONT DEVICE – OPTION IGNORED	A physical device specified on the :OPLBLT command must not be a symbiont device. PASS2 continues.
***xx ASSIGNMENT UNKNOWN – OPTION IGNORED	The name specified by xx is an unknown device type or logical device name. PASS2 continues.
***2 CHARS MAX FOR OPLABEL – OPTION IGNORED	The operational label must be one or two characters in length. PASS2 continues.
***xx OPLABEL IS TYPMNEMONIC/ LDEVICE – OPTION IGNORED	The operational label cannot be the same as a device type or a logical device name. PASS2 continues.
***xxxxx UNKNOWN DEVICE – OPTION IGNORED	The specification xxxxx is an unknown physical device. PASS2 continues.
MON Messages	
***AVGSER IS OUT OF RANGE – DEFAULT (1) IS USED	Self-explanatory. PASS2 continues.
***BIG9 SPECIFIED – SIG9 ALSO INCLUDED	When the BIG9 option is specified, the SIG9 option is assumed by default. PASS2 continues.
***ERROR IN PROCESSOR – JOB ABORTED	The processor is in need of repair. PASS2 error exits to the monitor.
***ILLEGAL TYPE OR SIZE	This message appears if a value was the wrong type (decimal, hexadecimal, or ndd) or was too large or too small, or if a text string contained too many characters. PASS2 ignores the value and continues with the next characters.
***INADEQUATE CORE SPACE – SKIP TO NEXT CC	The load module cannot be generated in the available core space.
***INVALID CHARACTER STRING	The length of the character string is either 0 or greater than 31, or the character string is not obtainable because of an error in trying to obtain a continuation record. PASS2 tries to continue.
***INVALID, UNKNOWN, OR DUPLICATE KEYWORD	A keyword was invalid, unknown, or duplicate. PASS2 searches for the next opening parenthesis '('.
***MODIFY ERROR – SKIP TO NEXT CC	The load module cannot be generated in the available core space.
***SYNTAX ERROR – 'X' EXPECTED	'X' is ')', ',', or '('. PASS2 searches for the next opening parenthesis.
***TROUBLE WITH SPEC:HAND – S9TRAPS &/OR ANSTP NOT INCLUDED	The BIG9 and/or SIG9 option was specified, but due to trouble with the SPEC:HAND file, the S9TRAPS module was not included or the ANSTP module (which was added if the ANSDET or ANSPROT option was specified) was not included. PASS2 tries to continue.

Table 73. PASS2 Messages (cont.)

Message	Description
COC Messages	
***COC0 IN NOT HIGHEST PRIORITY	COC0 input must have the highest priority of all COC interrupts. PASS2 discontinues processing of the :COC command.
***COCx COC INTERRUPT LEVEL CONFLICT – COC ABORTED	Self-explanatory. PASS2 skips to the next control command. M:COC is not generated.
***COCx DEVICE NOT DEFINED – COC ABORTED	The reference device has not been defined by a DEVICE control command.
***COCx DEVICE OPTION MISSING – COC ABORTED	Self-explanatory. PASS2 terminates processing of the :COC command. M:COC is not generated.
***COCx 'IN' NOT IN SAME GROUP AS OTHER INS	Self-explanatory. PASS2 discontinues processing of the :COC command.
***COCx LINES >64 – DEFAULT TAKEN	Self-explanatory. COC continues.
***COCx 'OUT' NOT IN SAME GROUP AS OTHER OUTS	Self-explanatory. PASS2 discontinues processing of the :COC command.
***COCx --RING>255 – 255 USED	The value for RING must be less than or equal to 255. COC continues.
***COCx RING INADEQUATE – DEFAULT TAKEN	Self-explanatory. COC continues.
***COCx WARNING: BUFFERS<3*LINES	The total number of buffers was less than three times the total number of lines. COC continues.
***TYPE>7 INVALID – DEFAULTS USED	Self-explanatory. COC continues.
IMC Messages	
***BPRIO<X'C0' OR>X'FF'. DEFAULT USED (X'FC')	Self-explanatory. PASS2 continues.
***COC BUFFERS>255 – 255 USED	Self-explanatory. IMC continues.
***EXPIRE OR MAXEXPIRE OUT OF RANGE – 999 DAYS, 23 HOURS USED	Self-explanatory. IMC continues.
***GPRIO<X'C0' OR>X'FF'. DEFAULT USED (X'FC')	Self-explanatory. PASS2 continues.
***INTIME OUT OF RANGE – DEFAULT (15) USED	Self-explanatory. PASS2 continues.
***LOGTIME OUT OF RANGE – DEFAULT (3) USED	Self-explanatory. PASS2 continues.
***MAXEXPIRE<EXPIRE – EXPIRE VALUE USED	Self-explanatory. IMC continues.
***MAXG<4 – DEFAULT (8) USED	Self-explanatory. IMC continues.
***MAXG>255 – DEFAULT (8) USED	Self-explanatory. IMC continues.

Table 73. PASS2 Messages (cont.)

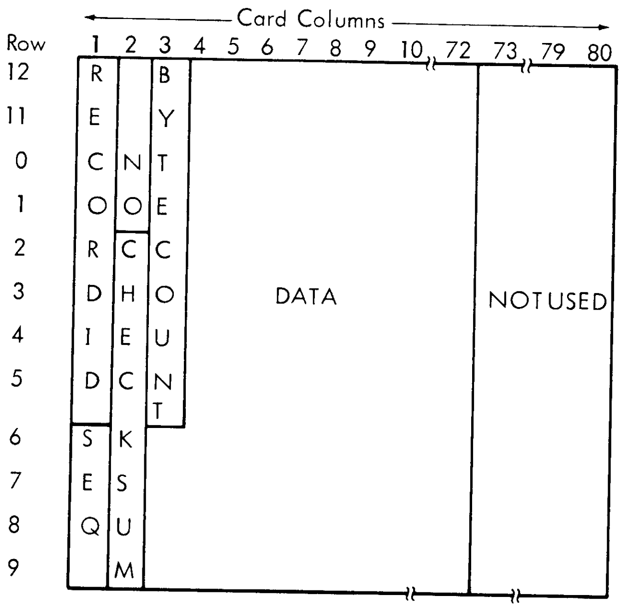
Messages	Description
***MINTIME >QUANTA – QUANTA VALUE USED	Self-explanatory. IMC continues.
***OPRIO<X'C0' OR>X'FF'. DEFAULT USED (X'FC')	Self-explanatory. PASS2 continues.
***RAMAX>63-READ AHEAD TABLES NOT GENERATED	Self-explanatory. PASS2 continues.
***RAMAX ZERO-RATO IGNORED	If RAMAX is zero, RATO is meaningless. PASS2 continues.
***RATO OUTSIDE LEGAL RANGE OF 0-32767, DEFAULT USED	Self-explanatory. (The default is 10000.) PASS2 continues.
***SWAPPER NOT DP – UCYL IGNORED	The UCYL option was specified for a RAD swap storage system. The option is ignored.
***TROUBLE WITH SPEC:HAND – RA NOT INCLUDED	An I/O error occurred when opening, reading, or writing the SPEC:HAND file.
***UCYL INVALID – DEFAULT (1) USED	The UCYL value was not 1 or 2. The default value 1 is used.
***UNBLOCK >BLOCK – SET EQUAL TO BLOCK	Self-explanatory. IMC continues.
***USERS >255. IMC ABORTED	Self-explanatory. IMC returns to P2CCI. M:IMC is not generated.
<u>SPROCS Messages</u> ***ATTEMPT TO REDEFINE xxxxxx. IGNORED	An attempt was made to redefine a standard processor without REPLACE. PASS2 continues.
***ILLEGAL FLAG f FOR nnnnnnn – FLAG IGNORED	This message appears if the flag "f" is specified for nnnnnnn and either is not an S, J, P, D, M, T, G, B, O, or C, or is a P, when nnnnnnn is not Pi. Only "f" is ignored, even when it occurs in a multi-flag option field. "f" is set to the offending character and nnnnnnn is set to the name.
***ILLEGAL TYPE OR SIZE	This message appears if a value was the wrong type (decimal, hexadecimal, or ndd) or was too large or too small, or if a text string contained too many characters. PASS2 ignores the value and continues with the next characters unless the illegal text string was to be a "name" for SPROCS or the next character should be a closing parenthesis, in which case PASS2 searches for the next opening parenthesis.
***INSUFFICIENT SPACE – SPROCS ABORTED	Too many overlays have been selected for the available core space to generate M:SPROCS.
***INVALID ALPHANUMERIC STRING	This message appears when a "name" or option (not "number") is not alphanumeric. The "name" or option is ignored and PASS2 continues with the next opening parenthesis or the next character, respectively.
***xxxxxxx NOT STANDARD PROCESSOR. CAN'T REPLACE	An attempt was made to replace a nonstandard processor. PASS2 continues.

Table 73. PASS2 Messages (cont.)

Message	Description
***PSPACE SIZE>S:CYLSZ – S:CYLSZ USED	PSA has been specified on a :DEVICE command defining a disk pack and the size parameter is not less than or equal to the number or granules per physical cylinder. It is set to an appropriate value obtained from S:CYLSZ (64 for the 7247 or 110 for the 7260/7265).
***TOO MANY VALUES	This message appears when one of the interim tables has been filled. Inadequate core space is implied; M:SPROCS is not generated. If the error position indicator ($) is under a CP-V overlay name, then the space required can be reduced by combining all flag options for each name into one multiflag option and replacing all "number" options with one "number" option equal to the sum of all the "number" options it replaces.
FRGD Messages ***NO CORE FOR FRGD, INTLB IGNORED	Self-explanatory. PASS2 continues.
***LABEL HAS ILLEGAL VALUE. IGNORED	The value falls outside of the three permissable ranges. FRGD continues.
***INTLB WITHOUT FRGD. OPTION IGNORED	Self-explanatory. PASS2 continues.
***TROUBLE WITH SPEC:HAND – RTROOT NOT INCLUDED	I/O error in opening, reading, or writing SPEC:HAND file.
XLIMIT Messages ***xxxx xxxxxx DEFAULT > MAX – DEFAULT SET TO MAX	This message applies to BLIMIT, OLIMIT, and GLIMIT commands and is self-explanatory. PASS2 continues.
***xxxxxxxx INVALID – SYSGEN DEFAULTS USED	The parameter identified by xxxxxxxx for an ELIMIT command was greater than X'7FFF'. The SYSGEN default was substituted. PASS2 continues.
XPART Messages **CONFLICT IN HOLD IN xxxxxxxx – PARTITION NOT HELD	Self-explanatory. XPART continues. (xxxxxxxx is the partition number.)
***CONFLICT IN LOCK IN xxxxxxxx – PARTITION NOT LOCKED	Self-explanatory. XPART continues.
***ERROR – PARAMETER *xxxx* IN xxxxxxxx MAX AND MIN INVALID – SYSGEN DEFAULTS USED	The maximum and minimum values are invalid. The default values are used. XPART continues.
***ERROR – PARAMETER *xxxx* IN xxxxxxxx MAX INVALID – DEFAULTS USED	The maximum value for option (either TIME or one of the resources) exceeds the permissible range. The default value is used. The default may be that specified prior to the first PART option or may be the standard SYSGEN default. XPART continues.
***ERROR – PARAMETER *xxxx* IN xxxxxxxx MAX< MIN – 0 USED FOR MIN	Option xxxx (either TIME or one of the resources) in either the default or identified partition number (xxxxxxxx) had a maximum value less than the minimum value. Zero was used for the minimum value. XPART continues.
***ERROR – PARAMETER *xxx* IN xxxxxxxx MIN INVALID – 0 USED	The minimum value for option (either TIME or one of the resources) exceeds the permissible range. Zero is used. XPART continues.
***PART 0 NOT ALLOWED – SPECIFIED RESOURCES IGNORED	Self-explanatory. XPART continues. Note: If PART > 16, the parameter is flagged as ILLEGAL TYPE OR SIZE and specified resources are also ignored.

LOCCT PROCESSOR

The LOCCT processor provides an optional phase of system generation that generates a file defining the elements and load structure of a user's processor or monitor. LOCCT may be called during system generation to create, in the current account, a permanent disk file containing the LOCCT, ROM, and TREE tables for a given set of LOAD (!LOCCT) and TREE control commands defining the load structure of a user's processor or monitor. It also outputs a copy of this file to the PO device. The contents of the LOCCT-generated file are referred to as "LOCCT tables". Each record consists of a binary card image having the format shown in Figure 93.



where

RECORD ID X'3E' binary card code and X'1E' binary end card code.

SEQ NO two-digit (hex) sequence number.

CHECKSUM byte checksum of card image.

BYTE COUNT number of useful bytes in card image, including control word in columns 1-3.

Figure 93. LOCCT Record Format

The LOCCT processor is entered via the following control command sequence:

!LOCCT(LMN,...)...(EF,(...))[†]
!TREE...
!DATA
:LOCCT name

where name specifies the name desired for use in retrieving the LOCCT file from disk. This name must be no longer than ten characters.

Continuation cards are not allowed. If comments are desired, they must be preceded by a period.

The "LOCCT name" command must immediately follow the !DATA command so that the Control Command Interpreter (CCI) will know that the LOCCT process is to be entered.

If the PASS3 processor is to be used to load a standard program at some later time, the LOCCT processor must be used once for every unique set of LOAD (!LOCCT command) and TREE control commands defining the load structure of a processor or monitor. It should be executed in the account in which PASS3 will eventually be executed. Also, all element file names in the LOAD (!LOCCT) command should originate in the account in which PASS3 will be executed. The file name used to generate the LOCCT file is determined by appending the "name" from the control command (see above) to the characters "LOCCT" (for the processor METASYM, the LOCCT file name would be LOCCTMETASYM). The name is optional. However, if the name does not appear, the M:EO DCB must have been previously assigned to the file name that PASS3 will reference later for a particular LOCCT. If a name appears, and an ASSIGN command assigns M:EO to a file name, the name on the LOCCT control command takes precedence. When the file name is determined via an ASSIGN command, the file name must include the first five characters (LOCCT) as part of the file name.

LOCCT EXAMPLES

Examples for using the LOCCT processor are shown below.

!JOB J1, LOAD ITEM, 1
!ASSIGN M:EO,(FILE, LOCCTXX)
!LOCCT(LMN,XX),...
!TREE
!DATA
:LOCCT
!Next monitor command

[†]This control command replaces the !LOAD command for this type of process and contains the same information the !LOAD command would normally contain.

This example will generate a permanent file, LOCCTXX, that will contain the LOCCT, ROM, and TREE tables for the first job's LOAD (!LOCCT) and TREE commands. The file will be under the account number "J1". A permanent copy will also be output to the PO device. This example's file name is determined by the "ASSIGN M:EO" control command.

```
!JOB J3,LOADITEM,1
!LOCCT (LMN,YYYY)...
!DATA
:LOCCT YYYY
!Next monitor command
```

This example will generate a permanent file, LOCCTYYYY, with the information from the LOAD (!LOCCT) command. The file will be under the account number "J3". A permanent copy will also be output to the PO device.

The example below will generate a permanent file, LOCCTZ using the information from the LOAD (!LOCCT command) and TREE commands. The file will be under the account number "J4". A permanent copy will also be output to the PO device. Notice that the ASSIGN command's file name is ignored and also that the LOCCT file name need not be the same as the load module name.

```
!JOB J4,LOADITEM,1
!ASSIGN M:EO,(FILE,LOCCTXYZ)
!LOCCT (LMN,XYZ)...
!TREE...
!DATA
:LOCCT Z
!Next monitor command
```

The permanent file name may be any name desired as long as LOCCT and PASS3 both reference the same name for a given LOCCT file.

LOCCT MESSAGES

Table 74 contains a list of error messages for LOCCT. All of these messages are output on the LL device.

Table 74. LOCCT Messages

Message	Description
***CANNOT GENERATE LOCCT WITH ROMS ON LABELED TAPE	An element file is on labeled tape.
***I/O ERR/ABN FOR READ C=xxxx	An I/O error or abnormal condition has been encountered on the C device. The value xxxx is the I/O error code.
***I/O ERR/ABN FOR WRITE EO=xxxx	An I/O error or abnormal condition has been encountered on the EO device. The value xxxx is the I/O error code.
***I/O ERR/ABN FOR WRITE PO=xxxx	An I/O error or abnormal condition has been encountered on the PO device. The value xxxx is the I/O error code.
LOCCT PROCESSOR ABORTED	This message is output after other LOCCT messages. LOCCT then exits to the monitor.
***NAME INVALID	The name in the LOCCT command was in error.
***NAME>10 CHARACTERS	The name in the LOCCT command was greater than ten characters in length.
***OPEN EO ERR/ABN=xxxx	An I/O error or abnormal condition has been encountered by LOCCT while trying to open the EO device. The value xxxx is the I/O error code.
***ROM TABLE END CANNOT BE FOUND	The ROM table is invalid.
***UNKNOWN CC OR CONTINUATION ILLEGAL	The name of the LOCCT command entered was invalid or the LOCCT command was to be continued. LOCCT displays the abort message and then exits to the monitor.

PASS3 PROCESSOR

The purpose of this processor is to cause the loading of standard monitors, processors, and libraries automatically via preestablished LOAD (!LOCCT) and TREE structures. These structures must have been generated previously by the LOCCT processor. PASS3 is entered via the control command

!PASS3 [option]

where the option is one of the following:

MON   specifies that PASS3 will abort if the monitor load is unsuccessful.

ALL   specifies that PASS3 will abort if any load is unsuccessful.

The commands that control PASS3 have the form

:id (option ,option)

where

id   is the name of a LOCCT information table that is to be obtained to define the load structure of a monitor, processor, or library subroutine (e.g., X, 9EDIT, FMGE). The id must not be longer than ten characters.

option   is optional information used to modify the default LOAD (!LOCCT) command structure in the LOCCT table, or may consist of general information to PASS3 (see the list of parameters below).

Any command that contains an asterisk in column 1 will be interpreted as a comment. Comments may also be added to any control command by preceding the comment with a period or semicolon. However, a semicolon requires a continuation.

A PASS3 control command identifies a standard system (monitor), processor, or library subroutine name for which a LOCCT table is to be obtained. This LOCCT table will describe to the loader how the named routine is to be loaded. The LOCCT will be assumed to be in the current account number, unless a previous ASSIGN command assigned the M:EI DCB to some other account number. PASS3 will accept LOCCT tables from only one account.

The control commands may contain optional parameters. The possible parameters are as follows:

SAVE($name_1$,$name_2$, ... ,$name_n$)[†]   specifies that the named element files will not be deleted from the disk. All options not saved will be deleted (see "DELETE" below).

DELETE[†]   specifies that when the loader has completed the loading of the standard system defined by id, all element files comprising this module will be deleted from the disk, except for those specifically saved (see "SAVE" above).

The deleted files must be in the current account, must not be protected by a password, and must be on disk rather than labeled tape. Unless saved, the LOCCT table will also be deleted.

When the LOCCT table for a particular id[††] has been obtained, PASS3 will obtain enough common storage to contain the LOCCT table and will save the LOCCT table in this area. This area is the communication area between PASS3 and the loader. PASS3 will then do an M:LINK call on the monitor requesting the loader (LOADER) as its overlay. The loader must appear as a load module file in the :SYS account. When the loader completes its function, it will do an M:LDTRC call on the monitor requesting a return to the calling parogram; namely, PASS3. At this time, the LOCCT table and all element files in the current account comprising the load module will be deleted from the disk if the DELETE and/or SAVE options were encountered in the control command. However, if the loader was unsuccessful in loading the element, the DELETE option will be ignored. PASS3 will then continue to its next control command.

All monitors should be loaded by PASS3 to take advantage of PASS3's ability to form the HANDLERS file automatically. If a monitor is to be loaded and there is no LOCCT for it, the LOAD (!LOCCT)/TREE commands must be used to generate a LOCCT by the LOCCT processor, and then PASS3 can use the LOCCT tables for the loading of the Monitor.

PASS3 EXAMPLES

!Previous monitor control command

!PASS3

:M:MON

:END

This example will obtain the LOCCT table in the file LOCCTM:MON. All element files comprising M:MON

[†]If the loader encounters an error during the loading of a module, the DELETE/SAVE feature will be ignored. That is, no element files will be deleted since the loader was not successful in loading them.

[††]The specified id is used to form the file name of a LOCCT table to be obtained. The id will be appended to the characters "LOCCT". This implies that for an id of FMGE, the LOCCT file name will be LOCCTFMGE.

will be saved. This load function might represent a minimal CP-V monitor (M:MON). The LOCCTM:MON file will be assumed to be in the account number under which this job is being run.

!Previous monitor control command

!PASS3

:CP7A

!Next monitor control command

This example is similar to the previous one, the file name being LOCCTCP7A. This load function might represent an alternate CP-V monitor system with load module name M:MON.

!JOB JX,LOADAUTO,1

!ASSIGN ME:EI,(FILE,DUMMY,ACCNT)

!PASS3

:PROC

:CCI

:LOADER (DELETE)

:Next control command

This example will obtain the LOCCT tables for LOCCTPROC, LOCCTCCI, and LOCCTLOADER from the ACCNT account and not the job's account, JX. All element files making up the LOADER will be deleted if it is loaded successfully.

PASS3 will obtain the information in SPEC:HAND, the file that SYSGEN PASS2 generated. The data in this file will identify which I/O handlers are required for this monitor (M:MON load module). If the SPEC:HAND file does not exist, PASS3 will abort the loading of the M:MON load module only, and will continue to the next LOCCT.

After obtaining the data from the SPEC:HAND file, PASS3 will open BASHANDL (the basic handler's file) as an input file and the file HANDLERS as an output file. The BASHANDL file will be copied to the new file HANDLERS and the BASHANDL file will be closed and saved. PASS3 will then obtain a handler name from the SPEC:HAND file, will open that file for input, and will copy the file to the new HANDLERS file. If a handler name from the SPEC:HAND file is found to be a part of the basic handlers file (BASHANDL), it will be ignored (see Table 75). When

Table 75. Handlers in BASHANDL File

Name	Device
KBTIO	TY
CRDIN	CR
PRTOUT	LP
PRTOUTL	LP
DISCIO	DC
DPAK	DP
DISKAB	DD

all of the handlers required have been merged into the HANDLERS file, this file will be closed and saved. All files accessed by PASS3 by this technique will be saved. PASS3 will then proceed to link to the overlay loader. If any handler cannot be found, PASS3 will abort the loading of the M:MON load module only, and will continue to the next LOCCT.

When PASS3 is being used to delete element files (DELETE option), certain restrictions must be adhered to. If a load module that is being formed has common element files with some other load module (e.g., PASS2 and PASS3 both contain MODIFY), and the DELETE option is desired, the load module must retain the common element until its final reference as an element file is encountered. The elements may be deleted as follows:

!JOB JZ,SPECIAL-DELETE-ROM,F

!PASS3

:PASS3 (SAVE(MODIFY))

:CCI (DELETE)

:PASS2 (DELETE)

This procedure will load PASS3 with DELETE (except for MODIFY), CCI with DELETE, and PASS2 with DELETE. Thus, PASS2 will result in deleting MODIFY. There are other combinations similar to this that incorporate other processors (Table 75).

PASS3 MESSAGES

Table 76 contains PASS3 messages. All of these messages are output on the LL device.

Table 76. PASS3 Messages

Message	Description
**BIN. CARD SEQUENCE ERR, SEQ. #xxxx	The LOCCT table contained a sequence number error. The value xxxx is the card sequence number where the error occurred. PASS3 continues to the next control command.
**BIN. CARD SEQUENCE TYPE, SEQ. #xxxx	The LOCCT table contained a sequence type error. The value xxxx is the card sequence number where the error occurred. PASS3 continues to the next control command.
**CANNOT OPEN/RELEASE	A SAVE or DELETE option was requested and a delete element could not be found. The file name that could not be found is output following this message. PASS3 continues to the next delete item.
**CC ERROR, NO ':' IN COLUMN 1	A PASS3 control command did not contain a ':' in column 1. PASS3 continues to the next control command.
**CC ID INVALID	The PASS3 control command's id is invalid. PASS3 continues to the next control command.
**CHECKSUM ERROR, SEQ. #xxxx	The LOCCT table contained a checksum error. The value xxxx is the card sequence number where the error occurred. PASS3 continues to the next control command.
**DELIMITER NOT (), = OR SYNTAX BAD	The syntax of a PASS3 command was either invalid or the expected delimiter should have been ( ), or =. PASS3 continues to the next control command.
**ID SIZE > 10 OR = 0 CHARACTERS	The id did not exist or was too large. PASS3 continues to the next control command.
**I/O ERR/ABN ON M:SI = xxxx	An I/O error or abnormal condition was encountered during I/O on the SI device. The value xxxx is the I/O error or abnormal code. PASS3 exits to the monitor.
**KEYWORD NOT DELETE/SAVE	A DELETE or SAVE keyword was expected but not found. PASS3 continues to the next control command.
**KEYWORD SAVE ALREADY USED	A second SAVE option was specified. PASS3 continues to the next control command.
**OPEN M:EI ERR/ABN = xxxx (LOCCT)	An I/O error or abnormal condition was encountered during an open operation on the EI device. The value xxxx is the I/O error or abnormal code. PASS3 then exits to the monitor.
**NAME INVALID	A name defined by the SAVE option was invalid. The name must be alphanumberic. PASS3 continues to the next name.
****OPEN/READ BASHANDL FILE ERR/ABN = xxxx	PASS3 could not obtain a given file when forming the HANDLERS file. The value xxxx is the error or abnormal code. BASHANDL names are replaced by the current file name being merged into the HANDLERS file. PASS3 continues to the next command.
****OPEN/READ SPEC:HAND FILE ERR/ABN = xxxx	PASS3 could not obtain a given file when forming the HANDLERS file. The value xxxx is the error or abnormal code. SPEC:HAND names are replaced by the current file name being merged into the HANDLERS file. PASS3 continues to the next command.

Table 76. PASS3 Messages (cont.)

Message	Description
####PASS3--COMPLETED####	PASS3 returned to the monitor.
####PASS3--IN--CONTROL####	PASS3 has been entered.
**READ M:EI ERR/ABN = xxxx (LOCCT)	An I/O error or abnormal condition was encountered during a read operation on the EI device. The value xxxx is the I/O error or abnormal code. PASS3 exits to the monitor.
****UNKNOWN TYPE CP USED	PASS3 type is unknown or not specified. PASS3 defaults to the running monitor type (CP-V).

DEF PROCESSOR

The DEF processor is called upon to generate PO tapes containing a bootable monitor system and all keyed files in the current account, and BO tapes containing the current monitor system from the :SYS account and all consecutive files in the current account.

The unlabeled portion of the tape includes any patches that can be read through M:PATCH (see Figure 91). If none can be read, the default patches '*' and '!' are used. The default assignment of M:PATCH is to the file PATCH in the running account, but it may be altered via an ASSIGN or SET command.

!DEF This command causes the monitor to fetch the DEF processor from the disk and to transfer control to it. The command has the form

!DEF [CP][,version#]

where

CP is the monitor type and need not be specified.

version# specifies a three-character field defining the version number of the target system. Only the first three characters specified are used.

The following commands are DEF control commands. Any control command that has an asterisk as its first character will be treated as a comment.

:INCLUDE This command allows the user to write to tape files that have a different organization than those automatically written (KEYED files to a PO tape, CONSEC files to a BO tape). The command has the form

:INCLUDE (item,item,...)

where item identifies a file to be included on the tape. Table 77 contains a list of files automatically INCLUDEd.

Multiple :INCLUDE commands and continued commands are allowed.

All INCLUDEd files are put in the :SYS account when the system is booted.

Table 77. Files Automatically INCLUDEd

PO Tape (from current account)	BO Tape (from :SYS account)
BPM	XDELTA
UTS	LOGON
SIG7FDP	TEL
FLIBMODE	SUPER
:BLIB	DEFCOM
SIGMET	SYMCON
M:CDCB	JIT0
M:OCDCB	ANLZ
M:BIDCB	CONTROL
M:CIDCB	ERRMSG
M:SIDCB	GHOST1
M:EIDCB	RECOVER
M:BODCB	M:SPROCS
M:CODCB	M:MON
M:SODCB	ALLOCAT
M:PODCB	CCI
M:GODCB	FILL
M:LODCB	LOADER

Table 77. Files Automatically INCLUDEd (cont.)

PO Tape (from current account)	BO Tape (from :SYS account)
M:DODCB	PASS2
M:EODCB	LOCCT
M:LLDCB	PASS3
M:SLDCB	DEF
M:ALDCB	PCL
M:LIDCB	
	plus each shared processor overlay as defined in M:SPROCS.

:IGNORE This command causes DEF to avoid writing to tape a file that would otherwise be written because of its organization. The command has the form

:IGNORE (item,item,...)

where item is a file to be ignored. (LASTLM and SPEC:HAND are automatically IGNOREd.)

The :IGNORE command does not override either stated or automatically INCLUDEd files. Multiple :IGNORE commands and continued commands are allowed.

:DELETE This command causes all files of KEYED organization for PO tapes or CONSEC organization for BO tapes as well as all IGNOREd files to be deleted from the current account as the tape is written. The command has the form

:DELETE

Multiple :DELETE commands are allowed.

The :INCLUDE, :IGNORE, and :DELETE commands apply until the next :WRITE is completed.

:WRITE This command causes DEF to write a tape. The command has the form

:WRITE [type][,outsn]

where

type specifies a PO or BO tape.

PO specifies that a tape is to be written with the bootable monitor portion obtained from the current account and all KEYED files obtained from the current account. PO is the default.

BO specifies that a tape is to be written with the bootable monitor portion obtained from the :SYS account and all CONSEC files obtained from the current account.

outsn specifies the serial number of the tape to be written. If no outsn is specified, the previous outsn for this DCB (M:PO or M:BO) is used again.

END This command causes DEF to exit.

If an end-of-file is encountered on M:SI input, one of two actions is taken. If the end-of-file follows a write command, DEF exits. Otherwise, DEF writes a PO tape and then exits.

Before DEF is called to write a tape, M:PO and/or M:BO should be ASSIGNed to a device and outsn.

DEF EXAMPLES

1. In the case below, DEF writes a normal CP-V PO tape.

   !ASSIGN M:PO,(DEVICE,9T),(SN,CPV1)

   !DEF CP,A01

   !Next monitor control command

2. In the following case. DEF writes LT#CPBO with FMGE among the included items and LT#CPPO with KEYED files K13 and K15 ignored.

   !ASSIGN M:PO,(DEVICE,9T),(OUTSN,CPPO)

   !ASSIGN M:BO,(DEVICE,MT),(OUTSN,CPBO)

   !DEF A01

   :INCLUDE (FMGE)

   :WRITE BO

   :IGNORE (K13,K15)

   :WRITE PO,A02

   !Next monitor control command

DEF MESSAGES

Table 78 contains DEF messages. All messages are output to the LL device.

Table 78. DEF Messages

Message	Description
****CANNOT OPEN OUTPUT DEVICE	DEF cannot open the output DCB. It proceeds to the next control command.
***CANNOT WRITE TAPE	DEF cannot write the tape. It releases the tape and goes to the next control command.
**CCTYPE UNKNOWN ****GET NEXT CC	An unknown control command was encountered. DEF continues.
**DELIMITER MUST BE ',' OR ')'	An invalid delimiter was encountered during the processing of an INCLUDE command. DEF continues to the next command.
***ILLEGAL INCLUDE – WILL BE COPIED LATER	An item on an :INCLUDE command has the wrong organization for :INCLUDE. DEF continues.
******INCLUDE FILE NOT FOUND	An INCLUDE item cannot be obtained. The INCLUDE item name is output preceding the message. DEF continues.
****INCLUDE ITEMS***	This is a subtitle message for the list of items included on the tape.
**NAME INVALID OR > 15 CHAR. LONG	An item name was either too large or not alphanumeric. DEF continues to the next name.
**NO ':' IN COLUMN-1	The control command did not contain a ':' in column 1. (Each control command, including a continuation command, must contain a ':' in column 1.) DEF continues to the next command.
****NOT ENOUGH CORE AVAILABLE ******SYSGEN DEF ABORTED	DEF did not have enough core for its use in processing the INCLUDE command and for writing a PO tape, or the DEF command did not define a valid type. DEF aborts.
OPENING LMname {RFDF, DATA, PRCD, DCBS, HEAD}	This message follows a standard system I/O error message and identifies what was being attempted when the error occurred. DEF aborts.
OTHER ITEMS	This is a subtitle message for the list of modules on the tape.
*****OUT OF ORDER PATCH	The patch listed above is out of order. DEF aborts.
*****OVERLAY EXCEEDS SIX PAGES	A monitor overlay is too large. DEF aborts.
.....nnn PATCHES INCLUDED	The number of patches that were included on the tape is specified by nnn.
.....PO TAPE CONTENTS..... .....BO TAPE CONTENTS.....	This is a title message for a list of the tape contents.
READING LMname {RFDF, DATA, PRCD, DCBS, HEAD}	This message follows a standard system I/O error message and identifies what was being attempted when the error occurred. DEF aborts.
ROOT SYMBOL TABLE:	This message precedes a list of symbols that may be used for patching.

Table 78. DEF Messages (cont.)

Message	Description
SYMBOLS FOR PATCHING, LMname	This message precedes a list of symbols that may be used for patching.
.....nn SYMBOLS MAY BE DEFINED	The number of new symbols that may be defined in the patch deck or on-line is specified by nn.
**SYNTAX ERROR, NO'('	An INCLUDE or IGNORE command did not define any items. DEF continues to the next command.
::::SYSGEN DEF IN CONTROL:::: ::::DEF COMPLETED::::	These messages are for title information only.
****TROUBLE WITH M:SPROCS	DEF cannot open or read M:SPROCS when writing BO. It proceeds to the next control command.
****WRITING PO BY DEFAULT	DEF is writing a PO tape because of an EOF or as the default of a :WRITE command.
*****XDELTA TOO BIG – REDUCE BIAS	The addition of symbol tables to XDELTA has made it larger than 64K. (See the Bootstrap and Patching Operations chapter.) DEF aborts. (XDELTA can be reloaded as low as E800 without code changes.)

ON-LINE SYSGEN

The PASS2, PASS3, DEF, and PCL processors can be run on-line as well as batch (LOCCT cannot). All of these processors read input through M:SI and print output through M:LL. M:SI may be assigned to a file for input and M:LL may be assigned to the line printer to avoid lengthy terminal listings.

When the processors are called on-line, they type

TYPE IS? $

The response should be (CR) or CP (CR) to specify CP-V. If DEF is called, the version number must follow the type, preceded by a comma. If PASS3 is called, the options must follow type, preceded by a comma.

The processors all prompt with a $ symbol. The END command will cause any processor to exit.

The input format is identical to that for batch and each line of input is sent back to the terminal unless M:LL is otherwise assigned.

Example:

```
XEROX CP-V AT YOUR SERVICE
ON AT 17:24 APR 15,'73
LOGON PLEASE: :SYSGEN,CP-V (RET)
!PASS2 (RET)
 ....PASS2 CCI IN CONTROL....
TYPE IS?   $ CP (RET)
$:CHAN (RET)
:CHAN
    :
    .
$END (RET)
 .....END OF PASS2.....
!SET M:SI DC/P3SI
        (P3SI is a file of PASS3 commands.)
!SET M:LL LP (RET)
        (Output is to the line printer.)
!PASS3 (RET)
OPTIONS (MON,ALL)?$MON (RET)
!SET M:SI 0 (RET)
        (Return SI to terminal.)
!SET M:PO 9T#CPPO (RET)
        (Assign device serial number.)
!DEF (RET)
TYPE IS? $ CP,A01 (RET)
$:INCLUDE (A1) (RET)
$:WRITE PO (RET)
$END (RET)
!
```

SYSGEN EXAMPLES

To simplify the system generation process, standard monitor systems and standard processors have been predefined in files on the CP-V release tapes. These files are described in the -11 release element.

17. BOOTSTRAP AND PATCHING OPERATIONS

SYSTEM TAPE FORMAT

A CP-V system tape contains the following elements:

1. Bootstrap loader.
2. Root for an absolute monitor.
3. General information record concerning this system tape.
4. Other monitor segments (XDELTA, ALLOCAT, GHOST1,).
5. Monitor overlay segments.
6. RECOVER.
7. Tape label information.
8. Files for all system load modules and other needed files.
9. Patches and GENMD commands.

The general arrangement of the information on a master system tape is shown in Figure 91.

PATCH DECK STRUCTURE

Patch decks have the following structure:

1. The following two types of patches (which may appear in any order):
   a. Delta format patches for the monitor root and its overlays.
   b. Symbol definition patches.
2. A card that contains only an asterisk (in column one). This card terminates the group of patches of type 1 above.
3. The following two types of patches (which may appear in any order):
   a. A GENDCB command to assign the account, a password, serial number, and type of tape drive for the boot tape.
   b. A group of GENMD commands and GENMD patches to the processors contained on the tape.
4. A !EOD command (the final command of the patch deck).

In addition, there are two types of cards that may appear anywhere within the patch deck (including within the GENMD portion). These two types are the conditional patch control command and the comment card.

No patch, command, or comment may contain more than 72 bytes of information.

The function and format of Delta format patches, symbol definition patches, GENDCB commands, GENMD commands, GENMD patches, conditional patch control commands, and comment cards are described in the paragraphs that follow.

DELTA FORMAT PATCHES

Delta format patches are used to patch various segments of the monitor. The format of a Delta format patch is:

[segname]/loc/value[(old value)]/comment

where

segname is the name of the segment to be patched. The current segnames and the order in which they must be patched is as follows:

ALLOCAT0	(ALLOCAT's DATA record)
ALLOCAT1	(ALLOCAT's PROCEDURE record)
GHOST10	(GHOST1's DATA record)
GHOST12	(GHOST1's DCBs record)
GHOST11	(GHOST1's PROCEDURE record)

.
.
.

Monitor overlay names in alphanumeric order

.
.
.

RECOVER

If a segname is present, the loc field must represent a location in the corresponding segment or the loc field (and value field) must be null. The latter type of patch would have the format

segname// (the third slash is unnecessary)

and must be the first patch with its particular segname.

Example:

```
OPEN// START THE OPEN PATCHES
OPEN/OPNH+.52/B PATCH/
/PATCH/LW,13 TABLES+.74/
```

If no segname is present, any location between 10_{16} and $FFP0_{16}$ may be patched. Such patches may appear anywhere within the patch deck.

loc is a Delta format symbolic location, possibly with offsets.

value is the Delta format value to be inserted at loc.

old value is the Delta format value of the previous contents of loc.

Example:

```
/IORT+.F8/PSM,9 TSTACK(PSM,6 TSTACK)/ FIX SIDR #6646
```

If a patch command is in error (e.g., has an illegal character, an incorrect old value, a value occupying more than one word, or an invalid loc value), it will be typed on the OC device. The operator must determine what was wrong and correct the problem.

If the error is apparent from examination of the patch, it can be corrected and the boot process restarted. If desired, the system may be examined with Executive Delta, which is now in control and requesting commands at the operator's console. The patch in error may be corrected from the operator's console using Delta by entering the patch correction mode by keying ¬ Ⓟ and then the correct patch in the form given above. After receiving the correct patch, the system resumes reading patches.

PATCH DECK SYMBOL TABLES

The Delta format symbolic values that are recognized in patches are assembled by DEF from the REF/DEF stacks of the patchable modules using these items:

1. All DSECT names.
2. All DEFs ending in a colon (with the colon removed).
3. The first UDEF after each CSECT unless a colon DEF intervened.
4. Patch segnames.

For M:MON only, all LDEFs are also included. The symbols obtained from M:MON and XDELTA are available to XDELTA at any time. Those from other modules are available only while that module is being patched. DEF lists the symbols that are included as the tables are created.

In addition, two special symbols are available during the patching process.

The first is the symbol @ whose value is equal to the next available location in the patch area of the monitor. That is, it is initially equal to the monitor symbol, MPATCH, and its value is incremented by one each time a patch is encountered whose loc field is equal to the current value of @. The use of the special symbol @ frees the user from having to allocate space in the PATCH area of the monitor since Executive Delta will not automatically relocate the patch area.

Example:

The following two patch decks are equivalent:

```
/IORT+.F8/B @/          /IORT+.F8/B PATCH/
/IORT+.FE/B @+1/        /IORT+.FE/B PATCH+1/
/@/LI,3 12/             /PATCH/LI,3 12/
/@/CB,3 5/              /.+1/CB,3 5/
/@/BNE .+2/             /.+1/BNE .+2/
/@/B IORT+.F9/          /.+1/B IORT+.F9/
/@/LI,3 0/              /.+1/LI,3 0/
/@/B IORT+.F9/          /.+1/B IORT+.F9/
```

The second special symbol is @@ and is used when an even address in MPATCH is required. The only restriction on this special symbol is that @@ cannot be referenced while patching @ (e.g., /@/@@/). The results are unpredictable.

New symbols may be added to the symbol table by including symbol definition patches in the patch deck. Symbol definition patches must have the format

#SYMBOL = value

where

SYMBOL is any Delta format symbol.

value is any evaluatable expression terminated by a blank.

Example:

```
#GRUNCH=.D87
/GRUNCH/B GRUNCH+.20/
/.+1/B @/
/@/LW,3 TABLES+3/
#JK=@
/@/CI,3 10/
/@/B GRUNCH+.50/
.
.
.
/55+.1E8/B JK/
```

In the above example, the patch at 55+.1E8 branches to the instruction CI,3 10.

GENDCB COMMAND

This command defines the system DCB associated with tape input during PASS0. This command is required only if the files are on a different tape than the boot tape or if they occupy more than one reel. If the command is not present in the patch deck, PASS0 reads the account and serial number from the tape and performs an automatic premount of the tape. No operator intervention is required.

The format of the GENDCB command is:

:GENDCB (M:BI,account[,password];

:,(INSN,value[,value]...),device)

where

M:BI specifies that tape input is to be through the M:BI DCB. No other DCB is valid for this command.

account specifies an account identifier (up to eight alphanumeric characters) associated with the labeled tape to be read during PASS0.

password is the password associated with the labeled tape to be read during PASS0. The password (if any) must correspond to that specified when the tape was created, and may be up to eight alphanumeric characters in length.

INSN,value,... specifies the serial number(s) (up to four alphanumeric characters in length) of the tape(s) to be read by PASS0. No more than three reels may be specified. The first reel specified must contain the first file to be read, and may be different from the reel used to boot the monitor.

device specifies a tape-type device code (e.g., 9T, 7T).

Example:

:GENDCB (M:BI,ACCT1,PASS1,;

:(INSN,001,002),9T)

Any number of GENDCB commands may appear in the patch deck. Only the last will be applied. If it is defective, files will be copied from the boot tape.

Any errors in the command are indicated by the message

***GENDCB ERROR

on the OC and LL devices.

GENMD COMMANDS

The GENMD commands are used in conjunction with the GENMD patches described below. The two GENMD commands are GENMD and LIST[D].

GENMD This command indicates which file is to be patched next. A GENMD command must precede the set of patches for each file to be patched. Any number of sets of patches to the same file may be present, provided each is preceded by a GENMD command. The format of the command is

GENMD filename

LIST[D] This command lists the patches currently in the file being patched and, if D is specified, deletes the record of them from the file. The format of the command is

LIST[D]

GENMD PATCHES

GENMD patches are used to modify nonresident elements of the system.

GENMD patches have the format:

$$\left\{ :\text{GENMD} \begin{matrix} [\text{,segname}] \\ [\text{segname}], \end{matrix} \right\} \text{loc,value[,value]...[.comment]}$$

where

segname specifies the overlay segment name to be patched. If not present, the most recently specified segname is assumed. If not present and no segname was specified previously, the root segname is assumed.

loc specifies the location to be patched and has the format name [± hex value]. The hexadecimal value is added to or subtracted from the absolute address of name. A maximum of eight characters may be used for the hexadecimal value. The name need not be defined in any particular overlay since all the stacks are searched. If more than one overlay defines the same name, the first is used.

value specifies the value to be inserted at loc. If more than one value is specified, they will be inserted at successive locations. Each value must have the format

hex value [±name[±name]...]

The absolute address of the names are added to or subtracted from the hexadecimal value. A maximum of eight characters may be used for the hexadecimal value. The name need not be defined in any particular overlay since all the stacks are searched. If more than one overlay defines the same name, the first is used.

A GENMD command may be continued by terminating the first line with a semicolon (;). The semicolon must not divide a name or a hexadecimal string and is not permitted where a blank is required. The continuation must begin with a colon (:) if the continued line began with :GENMD. Otherwise, the continuation begins with the next character of the command.

GENMD ERROR MESSAGES

Table 79 lists the error messages that may be output when GENMD commands and patches are being processed.

CONDITIONAL PATCH CONTROL COMMANDS

A conditional patch control command specifies whether the patches that follow are to be used as patches or are to be effectively ignored. The conditional patch control command controls the SKIP flag. When the SKIP flag is set, all subsequent patches are effectively ignored until the SKIP flag is reset. The conditional patch control command can appear any number of times and anywhere within the patch deck (including the GENMD portion). The command has the format:

#[value]

where value is any well-formed, but not necessarily evaluatable, expression terminated by a blank. The value expression may contain an undefined symbol.

Table 79. GENMD Error Messages

Message	Description
BAD LMN - 0000	The file is not a load module.
BAD LMN - xxxx	An error occurred when accessing the load module. The code and subcode are indicated by xxxx.
BAD SEG	A segname is not in the TREE.
DLM AT xx	The delimiter in column xx is not what it should be.
**nn GENMD ERRORS DETECTED	This message is output on the OC and LL devices at the conclusion of the GENMD patching process and indicates how many errors occurred.
HEX AT xx	The hexadecimal number ending in column xx is null, too large, or not hexadecimal.
LOC AT xx	The location ending in column xx or whose value ends in column xx is not contained in the segment.
NAME AT xx	The name ending in column xx is null or is not in the load module's stacks.
NO FILE NAMED	A 'GENMD filename' command has not yet been encountered or has no filename on it.
TOO BIG	Not enough core is available. It may be possible to do the patch if all names are converted to absolute hexadecimal values, since the stacks are read only if a name is used.

If value contains an undefined symbol, is negative, or is zero, the SKIP flag is set. While the SKIP flag is set, only the segname field of a patch is examined to determine when the current segment's patches end. If value is absent or greater than zero, the SKIP flag is reset and normal patching resumes.

The SKIP flag is also changed when a Delta format patch that does not have a loc and value field is encountered (i.e., segname//). In this case, it is set if the segname is undefined and it is reset otherwise.

Examples:

1. The following patches will be included only if the system was generated for a large Sigma 9:

```
#:9
/SWAPPER+.C5/B @/
/@/L1,5 0/
/@/SLS,7 L1/
/@/B SWAPPER+.C6/
#
```

2. The following patches will be included only if the ENQ/DEQ feature was included in the system:

```
ENQ//
ENQ/ENQO+.266/B @(CW,13 ENQP+.1F4)/
/@/LB,15 ENQP+.1F4/
/@/CB,15  13/
/@/B ENQO+.256/
OPEN//
```

COMMENT CARDS

Comment cards may appear anywhere within the patch deck. In the portion of the deck that contains Delta format patches and symbol definition patches, the comment card must contain a 'less than' character (<) in column one. In the GENDCB and GENMD portion of the deck, comment cards must contain one of the following in column one:

<

*

.

PATCH FILE CREATION

All patches read during the startup of the system (except GENDCB commands) are copied to the file PATCH in the system account. Those that came from the card reader appear in PATCH with the word CARD in columns 73-76. Those that were read while the skip flag was set appear with the word SKIP in columns 77-80. The resulting file may be used as input to DEF to create a system tape with the complete, current patch deck on it.

SEQUENCE OF OPERATIONS

The master system tape is loaded into the machine by use of the standard load procedure described in the CP-V/OPS Reference Manual, 90 16 75. The hardware bootstrap loads and enters the tape boot at the beginning of the system tape. This tape boot, in turn, loads the monitor root and enters at INITIAL.

The INITIAL routine performs a number of functions, as follows.

If the system was generated with the BIG 9 option on the :MON card and is not being booted on a Sigma 9, the following message is output and the bootstrap operation is terminated.

```
SYSTEM REQUIRES A SIGMA 9
```

Otherwise, the routine outputs the following message to the operator:

```
ENTER ANY OF:
I = TTY I/O
P = LP OUTPUT
F = TAPE FILES
T = TAPE PATCHES
C = CARD PATCHES
D = XDELTA
```

The operator must respond by typing one or more of the characters above followed by new line or by entering new line alone. If new line alone is entered, F and T are assumed by default. If any characters other than those listed above are entered, they are ignored.

The letters have the following meanings:

I specifies that the operator wants to read and respond to the normal OC messages during the boot. Otherwise, default responses are assumed up to the date/time request (see below) and normal output is suppressed. (Error messages will still be output.)

P causes output to the LL device to occur. Otherwise, the printer is not used.

F causes PASS0's tape copy operation to occur. Otherwise, a boot-under-the-files occurs.

T and C define the patching device(s). Either, neither, or both may be specified. If both are specified, cards will be read first for root patches and last for overlay patches. Card patches meant to repatch tape root patches should therefore be placed after a nonroot patch. Patches of the format segname// should be used in both patch decks to prevent the switching of devices from splitting up a logical patch.

D causes Executive Delta to be retained after the boot for debugging purposes. It also causes the subsequent question about Delta (see below) to be suppressed even if 'I' is specified. D must be specified if Executive Delta is to be retained and 'I' is not specified.

N is meaningful only by itself and means "none of the above".

If I was not specified, the next message to be output requests the date (see several paragraphs below).

If I was specified and if the system includes the real-time option, the system then issues the following message.

```
RESDF SIZE IN PAGES?
```

The message requests the operator to input the number of pages to be dedicated resident foreground pages. Any decimal value from 0 to 128 may be entered. If NEW LINE alone is entered, the SYSGEN-defined default for number of dedicated resident foreground pages will be used. In any case, the pages start at physical address 64K.

The system then outputs the message:

```
C/LL/COC/DC ASSIGN OK (YES/NO)
```

If the operator's response is YES or (NL), the subroutine assumes that the device addresses for the control device, listing log, and system device are not to be changed from those established when the monitor was defined. If the response is NO, then the following messages will be output to redefine these device addresses.

```
CRndd⇒CR
```

```
LPndd⇒LP
```

```
DCndd⇒DC
```

```
MEndd⇒ME
```

where each ndd is replaced by the current device and as many DC and ME messages are output as there are swap devices and COC devices, respectively.

In response to each of these messages the operator must type two or three characters. If two characters are typed, they must be 'SA' and indicate no change for this device. If three characters are typed, they must be the channel and device designation codes (ndd) defining the address of the indicated device (see Appendix C, Tables C-2 and C-3).

If the DC assignment is incorrect, the following message will request a new disk device address key-in:

```
DISC NOT 72x2
DCndd⇒DC
```

where x is replaced by 4, 3, or 1, whichever is required.

Before completing any of the above responses with a or (NL), the operator may cancel the response by striking the [illegible] key. Following this, or if a completed response is in error, the message

```
??
```

will be output and the key-in request will be repeated.

After all necessary responses have been received, the boot subroutine reads the system information record from tape and writes it on the LL and OC devices. Then it begins reading the patches.

When the first nonroot patch command is encountered, a subroutine at SWAPINIT reads and patches the OVERLAYs, ALLOCAT, GHOST1, and RECOVER. It then copies them to the swapping device, communicating the sizes and disk addresses to the resident root of the absolute monitor. Control then passes to another boot subroutine at WRTROOT. This second boot subroutine causes the monitor root to be copied to the disk, preceded by a disk bootstrap. At this point, the resident monitor is operational but the system has not yet been established on the resident swapping device. To establish this, the GHOST1 processor (PASS0, SYSMAK, and COCINIT) is logged on. GHOST1 first queries

```
DO YOU WANT DELTA (Y/N)?
```

to determine whether the system debugger's memory should be released. The operator has six seconds to respond after which N is assumed.

After PASS0 and SYSMAK have been executed, GHOST1 requests the current date and time key-ins with the following messages:

```
DATE (MM/DD/YY) =
```

```
TIME (HH:MM) =
```

The operator must respond by typing the appropriate quantity in the indicated format, terminating the response as usual with (NL), [illegible], or [illegible]. Then, after COCINIT is executed, GHOST1 will log-on the FILL ghost and delete itself.

BOOTING FROM DISK

Once the operating system has been bootstrapped from tape, it may thereafter be brought into core from the disk by means of the load procedure described in the CP-V/OPS Reference Manual, 90 16 75.

The hardware boot routine loads and transfers control to the disk boot which then loads the Monitor root into core and enters at INITIAL. The system is initialized, but GHOST1 skips PASS0 following a disk boot. The Monitor then enters the idle state and normal operation may be resumed.

BOOTSTRAP I/O ERROR RECOVERY

I/O error recovery during bootstrap is provided for the card reader, line printer, magnetic tape, and disk. However, error recovery is not possible until the tape boot and Monitor root have been read from tape. The following error messages may appear on the OC device:

xx INOPERATIVE

xx ERROR. TIO value TDV value

xx MANUAL MODE

CHECKWRITE ERROR

where

xx is MT, CR, DC, or LP.

value indicates the TIO or TDV results.

When either of the first two messages above occurs, the wait state is entered. To continue, the operator must place the CPU into IDLE, STEP, and then RUN state. The I/O will then be retried. If the third message above occurs, I/O will continue when the condition is corrected. When an error occurs for a magnetic tape or disk operation, the operation is retried ten times before an error message is output. If the fourth message above occurs, the wait state is entered. To continue, the operator must place the CPU into IDLE, STEP, and then RUN state. This message will appear if the checkwrite on disk fails. The checkwrite will be executed only if hardware sense switch 1 is set.

PASS0 PROCESSOR

The PASS0 processor performs various system initialization functions and constitutes a preliminary part of system generation, since it defines the environment in which system generation takes place. However, PASS0 is entered automatically whenever a CP-V tape is booted, regardless of whether or not system generation is to be done.

PASS0 reads a tape specified by the user (via the GENDCB command) which contains the nonresident elements of the system (i.e., CCI, processors, libraries, etc.). (This is normally the labeled portion of the tape used to bootstrap the absolute monitor.) PASS0 allows the user to modify these elements via the GENMD portion of the deck.

PASS0 MESSAGES

The messages in Table 80 may be output by the PASS0 program on the LL device. PASS0 continues its normal operation.

Table 80. PASS0 Messages

Message	Description
***CANNOT BOOT LMN	A load module cannot be read from the bootstrap tape because core is not large enough. PASS0 outputs the filename in error and continues to the next file, thus ignoring the file in error.
I/O ERR/ABN nn,xxxING FILE fffffff ON dddd	An I/O error or abnormal condition has occurred on tape or disk. nn is the error or abnormal code. xxxx is READ, WRITE, OPEN, or CLOSE. fffffff is the current filename. dddd is TAPE or DISC. PASS0 continues after this message.

APPENDIX A. XEROX STANDARD SYMBOLS, CODES AND CORRESPONDENCES

XEROX STANDARD SYMBOLS AND CODES

The symbols listed here include two types: graphic symbols and control characters. Graphic symbols are displayable and printable; control characters are not. Hybrids are SP (the symbol for a blank space), and DEL (the delete code) which is not considered a control command.

Two types of code are also shown: (1) the 8-bit Xerox Standard Computer Code, i.e., the Xerox Extended Binary-Coded-Interchange Code (EBCDIC); and (2) the 7-bit American National Standard Code for information Interchange (ANSCII), i.e., the Xerox Standard Communication Code.

XEROX STANDARD CHARACTER SETS

1. EBCDIC

   57-character set: uppercase letters, numerals, space, and & - / . < > ( ) + | $ * : ; , % # @ ' =

   63-character set: same as above plus ¢ ! _ ? " ¬

   89-character set: same as 63-character set plus lowercase letters

2. ANSCII

   64-character set: uppercase letters, numerals, space, and ! " $ % & ' ( ) * + , - . / \ ; : = < > ? @ _ [ ] ^ # | ¬

   95-character set: same as above plus lowercase letters and { } ¦ ~ `

CONTROL CODES

In addition to the standard character sets listed above, the Xerox symbol repertoire includes 37 control codes and the hybrid code DEL (hybrid code SP is considered part of all character sets). These are listed in the table titled CP-V Symbol-Code Correspondences.

SPECIAL CODE PROPERTIES

The following two properties of all Xerox standard codes will be retained for future standard code extensions:

1. All control codes, and only the control codes, have their two high-order bits equal to "00". DEL is not considered a control code.

2. No two graphic EBCDIC codes have their seven low-order bits equal.

Table A-1. CP-V 8-Bit Computer Codes (EBCDIC)

			Most Significant Digits															
	Hexadecimal		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
		Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Least Significant Digits	0	0000	NUL	DLE	LF only	ESC F	SP	&	-	∧					SP		-	0
	1	0001	SOH	X-ON	FS	CAN	[7]		/	¨	a	j		\ [1]	A	J		1
	2	0010	STX	DC2	GS	ESC X	⊥		⌈	¯	b	k	s	{ [1]	B	K	S	2
	3	0011	ETX	X-OFF	RS	ESC P		⎕			c	l	t	} [1]	C	L	T	3
	4	0100	EOT	DC4	US	ESC U	⌊		↓	≤	d	m	u	[[1]	D	M	U	4
	5	0101	HT	LF NL	EM	ESC (	∊	⊤			e	n	v	] [1]	E	N	V	5
	6	0110	ACK	SYN [8]	/	ESC)		∘	ω	≥	f	o	w		F	O	W	6
	7	0111	BEL	ETB	∧	ESC T			⊃		g	p	x		G	P	X	7
	8	1000	EOM BS	CAN	=	ESC S	Δ				h	q	y		H	Q	Y	8
	9	1001	ENQ	EM	CR only	ESC E	ι			∨	i	r	z		I	R	Z	9
	A	1010	NAK	SUB	EOT	ESC C	¢ [2]	!	^ [1]	:					[7]			×
	B	1011	VT	ESC	BS	ESC LF	.	$	,	#								÷
	C	1100	FF	FS	)	X-ON	<	*	%	@				[[6]				→
	D	1101	CR	GS	HT	X-OFF	(	)	_	'				] [6]				←
	E	1110	SO	RS	LF only	ESC R	+	;	>	=				Lost Data [6]				
	F	1111	SI	US	SUB	ESC CR	| [2]	¬ [2]	?	"			[6] |	¬ [6]				DEL
			3		6		4,7				5							

Notes:

1 The characters ^ \ { } [ ] are ANSCII characters that do not appear in any of the Xerox EBCDIC-based character sets, though they are shown in the EBCDIC table.

2 The characters ¢ | ¬ appear in the Xerox 63- and 89-character EBCDIC sets but not in either of the Xerox ANSCII-based sets. However, Xerox software translates the characters ¢ | ¬ into ANSCII characters as follows:

EBCDIC	=	ANSCII
¢		` (6-0)
|		¦ (7-12)
¬		~ (7-14)

3 The EBCDIC control codes in columns 0 and 1 and their binary representation are exactly the same as those in the ANSCII table, except for two interchanges: LF/NL with NAK, and HT with ENQ.

4 Characters enclosed in heavy lines are included only in the Xerox standard 63- and 89-character EBCDIC sets.

5 These characters are included only in the Xerox standard 89-character EBCDIC set.

6 The EBCDIC codes in column 3 are used by COC to perform special functions. The EBCDIC codes in column 2 and positions AF and BC through BF are used by COC for output only.

7 APL characters are assigned EBCDIC values that fall within the shaded area of the CP-V code set. These assignments are for APL internal use and are only reflected in 2741-APL translation tables.

8 Placing a SYN code as the last position of a nontransparent message will prevent the transmission of the SYN and the normal message appendage of the CR/LF pair. This allows a user to continue writing more than one message on the same line without affecting the carrier position. The EBCDIC SYN code is translated to an idle (IL) on output to 2741 terminals.

Table A-2. CP-V 7-Bit Communication Codes (ANSCII)

		Most Significant Digits							
Decimal (rows) (col's.)→		0	1	2	3	4	5	6	7
↓ Least Significant Digits	Binary	x000	x001	x010	x011	x100	x101	x110	x111
0	0000	NUL	DLE	SP	0	@	P	`	p
1	0001	SOH	DC1	! [5]	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
10	1010	LF NL	SUB	*	:	J	Z	j	z
11	1011	VT	ESC	+	;	K	[4] [[5]	k	{
12	1100	FF	FS	,	<	L	\	l	¦
13	1101	CR	GS	-	=	M	[4]] [5]	m	} [4]
14	1101	SO	RS	.	>	N	[4] ^ [5]	n	~ [4]
15	1111	SI	US	/	?	O	_ [4]	o	DEL
		2		3					

Notes:

1 Most significant bit, added for 8-bit format, is either 0 or an even-parity bit for the remaining 7 bits.

2 Columns 0-1 are control codes.

3 Columns 2-5 correspond to the Xerox 64-character ANSCII set.
Columns 2-7 correspond to the Xerox 95-character ANSCII set.

4 On many current teletypes, the symbol

^ is ↑ (5-14)
_ is ← (5-15)
~ is ESC or ALTMODE control (7-14)
} is ESC or ALTMODE control (7-13)

and none of the symbols appearing in columns 6-7 are provided. Except for the four symbol differences noted above, therefore, such teletypes provide all the characters in the Xerox 64-character ANSCII set. (The Xerox 7015 Remote Keyboard Printer provides the 64-character ANSCII set also, but prints ^ as Λ. It also interprets the [ ] characters as | ¬ .)

5 On the Xerox 7670 Remote Batch Terminal, the symbol

! is | (2-1)] is ! (5-13)
[is ¢ (5-11) ^ is ¬ (5-14)

and none of the symbols appearing in columns 6-7 are provided. Except for the four symbol differences noted above, therefore, this terminal provides all the characters in the Xerox 64-character ANSCII set.

Table A-3. CP-V Symbol-Code Correspondences

EBCDIC† Hex.	EBCDIC† Dec.	Symbol	Card Code	ANSCII††	Meaning	Remarks
00	0	NUL	12-0-9-8-1	0-0	null	**00 through 1F are control codes.**
01	1	SOH	12-9-1	0-1	start of header	**On 2741 terminals, SOH is PRE.**
02	2	STX	12-9-2	0-2	start of text	**On 2741 terminals, STX is BY.**
03	3	ETX	12-9-3	0-3	end of text	**On 2741 terminals, ETX is RES.**
04	4	EOT	12-9-4	0-4	end of transmission	
05	5	HT	12-9-5	0-9	horizontal tab	**00, 06, 07, 09-0B, and 0E-0F are idles for 2741 terminals.**
06	6	ACK	12-9-6	0-6	acknowledge (positive)	
07	7	BEL	12-9-7	0-7	bell	
08	8	BS or EOM	12-9-8	0-8	backspace or end of message	**EOM is used only on Xerox Keyboard/ Printers Models 7012, 7020, 8091, and 8092.**
09	9	ENQ	12-9-8-1	0-5	enquiry	
0A	10	NAK	12-9-8-2	1-5	negative acknowledge	
0B	11	VT	12-9-8-3	0-11	vertical tab	
0C	12	FF	12-9-8-4	0-12	form feed	
0D	13	CR	12-9-8-5	0-13	carriage return	CR outputs CR and LF.
0E	14	SO	12-9-8-6	0-14	shift out	
0F	15	SI	12-9-8-7	0-15	shift in	
10	16	DLE	12-11-9-8-1	1-0	data link escape	
11	17	DC1	11-9-1	1-1	device control 1	On Teletype terminals, DC1 is X-ON.
12	18	DC2	11-9-2	1-2	device control 2	On 2741 terminals, DC2 is PN.
13	19	DC3	11-9-3	1-3	device control 3	DC3 is RS on 2741s and X-OFF on Teletypes.
14	20	DC4	11-9-4	1-4	device control 4	On 2741 terminals, DC4 is PF.
15	21	LF or NL	11-9-5	0-10	line feed or new line	LF outputs CR and LF.
16	22	SYN	11-9-6	1-6	sync	
17	23	ETB	11-9-7	1-7	end of transmission block	On 2741 terminals, ETB is EOB.
18	24	CAN	11-9-8	1-8	cancel	
19	25	EM	11-9-8-1	1-9	end of medium	
1A	26	SUB	11-9-8-2	1-10	substitute	Replaces characters with parity error.
1B	27	ESC	11-9-8-3	1-11	escape	
1C	28	FS	11-9-8-4	1-12	file separator	
1D	29	GS	11-9-8-5	1-13	group separator	10, 11, 16, 18, 19, and 1B-1E are idles for 2741 terminals.
1E	30	RS	11-9-8-6	1-14	record separator	
1F	31	US	11-9-8-7	1-15	unit separator	
20	32	LF only	11-0-9-8-1	1-5	line feed only	20 through 2F are used by COC for output only. These codes are duplicates of the label entries that caused activation. The 20-2F entries output a single code only and are not affected by any special COC functional processing.
21	33	FS	0-9-1	1-12		
22	34	GS	0-9-2	1-13		
23	35	RS	0-9-3	1-14		
24	36	US	0-9-4	1-15		
25	37	EM	0-9-5	1-9		
26	38	/	0-9-6	2-15		
27	39	:	0-9-7	5-14		
28	40	=	0-9-8	3-13		
29	41	CR only	0-9-8-1	0-13	carriage return only	
2A	42	EOT	0-9-8-2	0-4		
2B	43	BS	0-9-8-3	0-8		
2C	44	)	0-9-8-4	2-9		
2D	45	HT	0-9-8-5	0-9	tab code only	
2E	46	LF only	0-9-8-6	1-5	line feed only	
2F	47	SUB	0-9-8-7	1-10		
30	48	ESC F	12-11-0-9-8-1		end of file	30 through 3F cause COC to perform special functions.
31	49	CANCEL	9-1		delete all input and output	
32	50	ESC X	9-2		delete input line	
33	51	ESC P	9-3		toggle half-duplex paper tape mode	
34	52	ESC U	9-4		toggle restrict upper case	
35	53	ESC (	9-5		upper case shift	
36	54	ESC)	9-6		lower case shift	
37	55	ESC T	9-7		toggle tab simulation mode	
38	56	ESC S	9-8		toggle space insertion mode	
39	57	ESC E	9-8-1		toggle echo mode	
3A	58	ESC C	9-8-2		toggle tab relative mode	
3B	59	ESC LF	9-8-3		line continuation	3B toggles the backspace edit mode for 2741 terminals.
3C	60	X-ON	9-8-4		start paper tape	
3D	61	X-OFF	9-8-5		stop paper tape	
3E	62	ESC R	9-8-6		retype	
3F	63	ESC CR	9-8-7		line continuation	

†Hexadecimal and decimal notation.

††Decimal notation (column-row).

Table A-3. CP-V Symbol-Code Correspondences (cont.)

EBCDIC† Hex.	EBCDIC† Dec.	Symbol	Card Code	ANSCII††	Meaning	Remarks
40	64	SP	blank	2-0	blank	
41	65		12-0-9-1			41, 43, 46, and 47 are unassigned.
42	66	⊥	12-0-9-2		decode	
43	67		12-0-9-3			
44	68	⌊	12-0-9-4		minimum	42, 44, 45, 48, and 49 are APL characters for 2741 APL use only.
45	69	ϵ	12-0-9-5		epsilon	
46	70		12-0-9-6			
47	71		12-0-9-7			
48	72	Δ	12-0-9-8		delta	
49	73	⍳	12-8-1		index	
4A	74	¢ or `	12-8-2	6-0	cent or accent grave	Accent grave used for left single quote. On Model 7670, ` not available, and ¢ = ANSCII 5-11. On 2741 APL, ¢ is ⊂ (subset).
4B	75	.	12-8-3	2-14	period	
4C	76	<	12-8-4	3-12	less than	
4D	77	(	12-8-5	2-8	left parenthesis	
4E	78	+	12-8-6	2-11	plus	
4F	79	ǀ or ¦	12-8-7	7-12	vertical bar or broken bar	On Model 7670, ¦ not available, and ǀ = ANSCII 2-1.
50	80	&	12	2-6	ampersand	On 2741 APL, & is ∩ (intersection). 51, 52, 54, 57, 58, and 59 are unassigned. 53, 55, and 56 are APL characters for 2741 APL use only.
51	81		12-11-9-1			
52	82		12-11-9-2			
53	83	⎕	12-11-9-3		quad	
54	84		12-11-9-4			
55	85	⊤	12-11-9-5		encode	
56	86	○	12-11-9-6		circular	
57	87		12-11-9-7			
58	88		12-11-9-8			
59	89		11-8-1			
5A	90	!	11-8-2	2-1	exclamation point	On Model 7670, ! is ǀ. On 2741 APL, ! is ° (degree). On 2741 APL, $ is ∪ (union).
5B	91	$	11-8-3	2-4	dollars	
5C	92	*	11-8-4	2-10	asterisk	
5D	93	)	11-8-5	2-9	right parenthesis	
5E	94	;	11-8-6	3-11	semicolon	
5F	95	~ or ¬	11-8-7	7-14	tilde or logical not	On Model 7670, ~ is not available, and ¬ = ANSCII 5-14.
60	96	-	11	2-13	minus, dash, hyphen	
61	97	/	0-1	2-15	slash	
62	98	⌈	11-0-9-2		maximum	62, 64, 66, and 67 are APL characters for 2741 APL use only.
63	99		11-0-9-3			
64	100	↓	11-0-9-4		down arrow	
65	101		11-0-9-5			
66	102	⍵	11-0-9-6		omega	63, 65, 68, and 69 are unassigned.
67	103	⊃	11-0-9-7		superset	
68	104		11-0-9-8			
69	105		0-8-1			
6A	106	^	12-11	5-14	circumflex	On Model 7670 ^ is ¬. On Model 7015 ^ is ∧ (caret). On 2741 APL, ^ is ↑. On 2741 APL, % is ⍴. Underline is sometimes called "break character"; may be printed along bottom of character line.
6B	107	,	0-8-3	2-12	comma	
6C	108	%	0-8-4	2-5	percent	
6D	109	_	0-8-5	5-15	underline	
6E	110	>	0-8-6	3-14	greater than	
6F	111	?	0-8-7	3-15	question mark	
70	112	∧	12-11-0		APL	70-72, 74, 76, and 79 are APL characters for 2741 APL use only.
71	113	¨	12-11-0-9-1		APL quote mark	
72	114	¯	12-11-0-9-2		overscore	
73	115		12-11-0-9-3			
74	116	≤	12-11-0-9-4		less than or equal	73, 75, 77, and 78 are unassigned.
75	117		12-11-0-9-5			
76	118	≥	12-11-0-9-6		greater than or equal	
77	119		12-11-0-9-7			
78	120		12-11-0-9-8			
79	121	∨	8-1		down delta	
7A	122	:	8-2	3-10	colon	
7B	123	#	8-3	2-3	number	
7C	124	@	8-4	4-0	at	
7D	125	'	8-5	2-7	apostrophe (right single quote)	
7E	126	=	8-6	3-13	equals	
7F	127	"	8-7	2-2	quotation mark	

† Hexadecimal and decimal notation.

†† Decimal notation (column-row).

Table A-3. CP-V Symbol-Code Correspondences (cont.)

EBCDIC† Hex.	Dec.	Symbol	Card Code	ANSCII††	Meaning	Remarks
80	128		12-0-8-1			80 is unassigned.
81	129	a	12-0-1	6-1		81-89, 91-99, A2-A9 comprise the lowercase alphabet. Available only in Xerox standard 89- and 95-character sets.
82	130	b	12-0-2	6-2		
83	131	c	12-0-3	6-3		
84	132	d	12-0-4	6-4		
85	133	e	12-0-5	6-5		
86	134	f	12-0-6	6-6		
87	135	g	12-0-7	6-7		
88	136	h	12-0-8	6-8		
89	137	i	12-0-9	6-9		
8A	138		12-0-8-2			8A through 90 are unassigned.
8B	139		12-0-8-3			
8C	140		12-0-8-4			
8D	141		12-0-8-5			
8E	142		12-0-8-6			
8F	143		12-0-8-7			
90	144		12-11-8-1			
91	145	j	12-11-1	6-10		
92	146	k	12-11-2	6-11		
93	147	l	12-11-3	6-12		
94	148	m	12-11-4	6-13		
95	149	n	12-11-5	6-14		
96	150	o	12-11-6	6-15		
97	151	p	12-11-7	7-0		
98	152	q	12-11-8	7-1		
99	153	r	12-11-9	7-2		
9A	154		12-11-8-2			9A through A1 are unassigned.
9B	155		12-11-8-3			
9C	156		12-11-8-4			
9D	157		12-11-8-5			
9E	158		12-11-8-6			
9F	159		12-11-8-7			
A0	160		11-0-8-1			
A1	161		11-0-1			
A2	162	s	11-0-2	7-3		
A3	163	t	11-0-3	7-4		
A4	164	u	11-0-4	7-5		
A5	165	v	11-0-5	7-6		
A6	166	w	11-0-6	7-7		
A7	167	x	11-0-7	7-8		
A8	168	y	11-0-8	7-9		
A9	169	z	11-0-9	7-10		
AA	170		11-0-8-2			AA through AE are unassigned.
AB	171		11-0-8-3			
AC	172		11-0-8-4			
AD	173		11-0-8-5			
AE	174		11-0-8-6			
AF	175	|	11-0-8-7		logical and	AF is used by COC for output of an ANSCII 7-12 code only.
B0	176		12-11-0-8-1			
B1	177	\	12-11-0-1	5-12	backslash	
B2	178	{	12-11-0-2	7-11	left brace	On 2741 terminals, { is output as (.
B3	179	}	12-11-0-3	7-13	right brace	On 2741 terminals, } is output as).
B4	180	[	12-11-0-4	5-11	left bracket	On Model 7670, [is ¢. On Model 7015, [is !.
B5	181	]	12-11-0-5	5-13	right bracket	On Model 7670,] is !. On Model 7015,] is ¬.
B6	182		12-11-0-6			80 and B6 through BB are unassigned.
B7	183		12-11-0-7			
B8	184		12-11-0-8			
B9	185		12-11-0-9			
BA	186		12-11-0-8-2			
BB	187		12-11-0-8-3			
BC	188	[	12-11-0-8-4		left bracket	BC, BD, and BF are used by COC for output of ANSCII 5-11, 5-13, and 7-14, respectively.
BD	189	]	12-11-0-8-5		right bracket	
BE	190	lost data	12-11-0-8-5		lost data	
BF	191	¬	12-11-0-8-7		logical not	On 2741 Selectric and EBCD Standard Keyboards, [is output as (and] is output as).

†Hexadecimal and decimal notation.

††Decimal notation (column-row).

Table A-3. CP-V Symbol-Code Correspondences (cont.)

EBCDIC†						
Hex.	Dec.	Symbol	Card Code	ANSCII††	Meaning	Remarks
C0	192	SP	12-0	2-0	blank	Output only.
C1	193	A	12-1	4-1		C1-C9, D1-D9, E2-E9 comprise the uppercase alphabet.
C2	194	B	12-2	4-2		
C3	195	C	12-3	4-3		
C4	196	D	12-4	4-4		
C5	197	E	12-5	4-5		
C6	198	F	12-6	4-6		
C7	199	G	12-7	4-7		
C8	200	H	12-8	4-8		
C9	201	I	12-9	4-9		
CA	202		12-0-9-8-2			CA through CF are unassigned.
CB	203		12-0-9-8-3			
CC	204		12-0-9-8-4			
CD	205		12-0-9-8-5			
CE	206		12-0-9-8-6			
CF	207		12-0-9-8-7			
D0	208		11-0			D0 is unassigned.
D1	209	J	11-1	4-10		
D2	210	K	11-2	4-11		
D3	211	L	11-3	4-12		
D4	212	M	11-4	4-13		
D5	213	N	11-5	4-14		
D6	214	O	11-6	4-15		
D7	215	P	11-7	5-0		
D8	216	Q	11-8	5-1		
D9	217	R	11-9	5-2		
DA	218		12-11-9-8-2			DA through DF are unassigned.
DB	219		12-11-9-8-3			
DC	220		12-11-9-8-4			
DD	221		12-11-9-8-5			
DE	222		12-11-9-8-6			
DF	223		12-11-9-8-7			
E0	224	–	0-8-2	2-13	minus	Output only. E1 is unassigned.
E1	225		11-0-9-1			
E2	226	S	0-2	5-3		
E3	227	T	0-3	5-4		
E4	228	U	0-4	5-5		
E5	229	V	0-5	5-6		
E6	230	W	0-6	5-7		
E7	231	X	0-7	5-8		
E8	232	Y	0-8	5-9		
E9	233	Z	0-9	5-10		
EA	234		11-0-9-8-2			EA through EF are unassigned.
EB	235		11-0-9-8-3			
EC	236		11-0-9-8-4			
ED	237		11-0-9-8-5			
EE	238		11-0-9-8-6			
EF	239		11-0-9-8-7			
F0	240	0	0	3-0		
F1	241	1	1	3-1		
F2	242	2	2	3-2		
F3	243	3	3	3-3		
F4	244	4	4	3-4		
F5	245	5	5	3-5		
F6	246	6	6	3-6		
F7	247	7	7	3-7		
F8	248	8	8	3-8		
F9	249	9	9	3-9		
FA	250	×	12-11-0-9-8-2		multiply	FA through FF are APL characters for 2741 APL use only.
FB	251	÷	12-11-0-9-8-3		divide	
FC	252	→	12-11-0-9-8-4		right arrow	
FD	253	←	12-11-0-9-8-5		left arrow	
FE	254		12-11-0-9-8-6			FE is not assigned.
FF	255	DEL	12-11-0-9-8-7		delete	Special – neither graphic nor control symbol.

†Hexadecimal and decimal notation.

††Decimal notation (column–row).

Table A-4. ANSCII Control-Character Translation Table

Input					Output	
ANSCII	TTY Key	Echoed	Prog. Receives (EBCDIC)	Process	EBCDIC	Transmitted (ANSCII)
NUL (00)	P^{cs}	None	None	None	NUL (00)	Nothing (end of output message)
SOH (01)†	A^c	SOH	SOH	None	SOH (01)	SOH
STX (02)†	B^c	STX	STX	None	STX (02)	STX
ETX (03)†	C^c	ETX	ETX	None	ETX (03)	ETX
EOT (04)†	D^c	EOT	EOT	Input Complete.	EOT (04)	EOT
ENQ (05)†	E^c	ENQ	ENQ (09)	None	HT (05)	Space(s) if tab simulation on, or HT (09) if not.
ACK (06)†	F^c	ACK	ACK	None	ACK (06)	ACK
BEL (07)	G^c	BEL	BEL	None	BEL (07)	BEL
BS (08)	H^c	BS	BS	None	BS (08)	BS
HT (09)	I^c	Space to tab stop if tab simulation on, or 1 space if not.	Spaces to tab stop, or one space, or tab (05) depending on space insertion mode.	None	ENQ (09)	ENQ (05)
LF/NL (0A)	NL	CR and LF	LF (15)	Input Complete.	NAK (0A)	NAK (15)
VT (0B)	K^c	VT	VT	None	VT (0B)	VT
FF (0C)	L^c	None	FF	Page Header and Input Complete.	FF (0C)	Page Header
CR (0D)	CR	CR and LF	CR (0D)	Input Complete.	CR (0D)	CR and LF (0A)
SO (0E)	N^c	SO	SO	None	SO (0E)	SO
SI (0F)	O^c	SI	SI	None	SI (0F)	SI
DLE (10)†	P^c	DLE	DLE	None	DLE (10)	DLE
DC1 (11)	Q^c	DC1	None	Paper Tape On.	DC1 (11)	DC1
DC2 (12)	R^c	DC2	DC2	None	DC2 (12)	DC2
DC3 (13)	S^c	DC3	None	Paper Tape Off.	DC3 (13)	DC3
DC4 (14)†	T^c	DC4	DC4	None	DC4 (14)	DC4
NAK (15)†	U^c	NAK	NAK (0A)	None	LF/NL (15)	CR and LF (0A)

†These characters are communication control characters reserved for use by hardware. Any other use of them risks incompatibility with future hardware developments and is done so by the user at his own risk.

Table A-4. ANSCII Control-Character Translation Table (cont.)

Input					Output	
ANSCII	TTY Key	Echoed	Prog. Receives (EBCDIC)	Process	EBCDIC	Transmitted (ANSCII)
SYN (16)†	V^c	SYN	SYN	None	SYN† (16)	SYN (not transmitted for last character in user's buffer).
ETB (17)†	W^c	ETB	ETB	None	ETB (17)	ETB
CAN (18)	X^c	Back-arrow and CR/LF	None	Cancel input or output message.	CAN (18)	CAN
EM (19)	Y^c	Back-arrow and CR/LF	None	Monitor Escape/ Control to TEL	EM (19)	EM
SUB (1A)	Z^c	SUB	SUB	Input Complete	SUB (1A)	# (A3)
ESC (1B)	K^{cs} ESC PREFIX	None	None	Initiate escape sequence mode.	ESC (1B)	ESC
FS (1C)	L^{cs}	FS	FS	Input Complete	FS (1C)	FS
GS (1D)	M^{cs}	GS	GS	Input Complete	GS (1D)	GS
RS (1E)	N^{cs}	RS	RS	Input Complete	RS (1E)	RS
US (1F)	O^{cs}	US	US	Input Complete	US (1F)	US
} (7D)	ALT-MODE	} or None	} or None	} if model 37; as ESC if model 33, 35, or 7015.	} (B3)	} (7D)
~(7E)	ESC (7015)	~or None	~or None	~if model 37; as ESC if model 33, 35, or 7015	¬ (5F)	~(7E)
DEL (7F)	Rubout	\	None	Rubout last character.	DEL (FF)	None
All ANSCII upper and lower case alphabetics are translated on input into the corresponding EBCDIC graphics as shown in Tables C-1 and C-2. All special graphics map as shown, allowing for Table C-1, Note 2, and the exceptions above for model 33 and 35. Lower case alphabetics map into corresponding EBCDIC upper case if the ESC U mode is set. Upper case alphabetics map into corresponding EBCDIC lower case if ESC) is set.					Alphabetic and symbol output translation is also as shown in Tables C-1 and C-2; for Models 33 and 35, and 7015 terminals, however, lowercase alphabetics are automatically translated to upper case.	

†These characters are communication control characters reserved for use by hardware. Any other use of them risks incompatibility with future hardware developments and is done so by the user at his own risk.

Table A-5. Substitutions for Nonexistent Characters on 2741 Keyboards

EBCDIC Character	APL Keyboard	Selectric Keyboard	EBCD Keyboard
>	>	, (upper case)	>
<	<	. (upper case)	<
^	↑	¢	¢
$\vert$	$\vert$	° (degree)	$\vert$
¬	~	±	¬
#	≠	#	#
%	ρ	%	%
¢	⊂	¢	¢
@	α	@	@
"	∇	"	"
!	∘	!	!
&	∩	&	&
$	∪	$	$

APPENDIX B. OPERATIONAL LABELS

Table B-1. Standard Operational Labels and Default Device Assignments

Operational Label	Batch Device	On-Line Device	Ghost Device
C	Card reader	Terminal	Operator's console
OC	Operator's console	Terminal	Operator's console
LO	Line printer	Terminal	Line printer
LL	Line printer	Terminal	Line printer
DO	Line printer	Terminal	Line printer
PO	Card punch	None	Card punch
BO	Card punch	None	Card punch
LI	Card reader	None	Operator's console
SI	Card reader	Terminal	Operator's console
BI	Card reader	None	Operator's console
SL	Line printer	Terminal	Line printer
SO	Card punch	None	Card punch
CI	Card reader	None	Operator's console
CO	Card punch	None	Card punch
AL	Card punch	None	Card punch
EI	Card reader	Terminal	Operator's console
EO	Card punch	None	Card punch
UC	Operator's console	Terminal	Operator's console

Table B-2. Batch Assignment of Operational Labels

Device	Oplabel
Line printer	LO, LL, DO, SL, LP
Card reader	C, LI, SI, BI, CI, EI, CR
Card punch	PO, BO, SO, CO, AL, EO, CP
Operator's console	OC, UC
9-track magnetic tape	9T
7-track magnetic tape	7T
Default tape	MT
None	NO, ME

Table B-3. On-Line Assignment of Operational Labels

Device	Oplabel
User's terminal	C, OC, LO, LL, DO, SI, SL, EI, UC, ME, CR
Card punch	CP
Line printer	LP
9-track magnetic tape	9T
7-track magnetic tape	7T
Default tape	MT
None	NO, PO, BO, LI, BI, SO, CI, CO, AL, EO, PR, PP

APPENDIX C. PHYSICAL DEVICE NAMES

A physical device name is indicated by yyndd.

where

yy specifies the type of device (see Table C-1).

n specifies the channel letter (see Table C-2).

dd specifies the device number (see Table C-3), in hexadecimal.

Table C-1. Standard I/O Device Type Codes

yy	Device Type
7T	7-track magnetic tape
9T	9-track magnetic tape
CP	Card punch
CR	Card reader
TY	Typewriter
LP	Line printer
DP	Disk pack
DC	Magnetic disk
ME	CP-V terminal
RB	Remote processing data set controller

Table C-2. Channel Designation Codes

Specified Channel Letter (n)	Corresponding Decimal Digit of Unit Address
A	0
B	1
C	2
D	3
E	4
F	5
G	6
H	7

Table C-3. Device Designation Codes

Hexadecimal Code (dd)	Device Designation
$00 \leq dd \leq 7F$	Refers to a device number (00 through 7F).
$80 \leq dd \leq FF$	Refers to a device controller number (8 through F followed by a device number 9 through F).

APPENDIX D. SIGMA STANDARD OBJECT LANGUAGE

INTRODUCTION

GENERAL

The Xerox Sigma standard object language provides a means of expressing the output of any Sigma processor in standard format. All programs and subprograms in this object format can be loaded by the Monitor's relocating loader.[†] Such a loader is capable of providing the program linkages needed to form an executable program in core storage. The object language is designed to be both computer-independent and medium-independent; i.e., it is applicable to any Xerox Sigma computer having a 32-bit word length, and the same format is used for both cards and paper tape.

SOURCE CODE TRANSLATION

Before a program can be executed by the computer, it must be translated from symbolic form to binary data words and machine instructions. The primary stages of source program translation are accomplished by a processor. However, under certain circumstances, the processor may not be able to translate the entire source program directly into machine language form.

If a source program contains symbolic forward references, a single-pass processor such as the Xerox Symbol assembler can not resolve such references into machine language. This is because the machine language value for the referenced symbol is not established by a one-pass processor until after the statement containing the forward reference has been processed.

A two-pass processor, such as the Xerox Meta-Symbol assembler, is capable of making "retroactive" changes in the object program before the object code is output. Therefore, a two-pass processor does not have to output any special object codes for forward references. An example of a forward reference in a Symbol source program is given below.

```
        :
Y       EQU     $ + 3
        :
        CI, 5   Z
        :
        LI, R   Z
        :
Z       EQU     2
        :
        BG      Z
        :
R       EQU     Z + 1
        :
```

In this example the operand $ + 3 is not a forward reference because the assembler can evaluate it when processing the source statement in which it appears. However, the operand Z in the statement

```
CI,5    Z
```

is a forward reference because it appears before Z has been defined. In processing the statement, the assembler outputs the machine-language code for CI,5, assigns a forward reference number (e.g., 12) to the symbol Z, and outputs that forward reference number. The forward reference number and the symbol Z are also retained in the assembler's symbol table.

When the assembler processes the source statement

```
LI,R    Z
```

it outputs the machine-language code for LI, assigns a forward reference number (e.g., 18) to the symbol R, outputs that number, and again outputs forward reference number 12 for symbol Z.

On processing the source statement

```
Z    EQU    2
```

the assembler again outputs symbol Z's forward reference number and also outputs the value, which defines symbol Z, so that the relocating loader will be able to satisfy references to Z in statements CI,5 Z and LI,R Z. At this time, symbol Z's forward reference number (i.e., 12) may be deleted from the assembler's symbol table and the defined value of Z equated with the symbol Z (in the symbol table). Then, subsequent references to Z, as in source statement

```
BG    Z
```

would not constitute forward references, since the assembler could resolve them immediately by consulting its symbol table.

If a program contains symbolic references to externally defined symbols in one or more separately processed subprograms or library routines, the processor will be unable to generate the necessary program linkages.

An example of an external reference in a Symbol source program is shown below.

```
REF     ALPH
:
LI,3    ALPH
:
```

When the assembler processes the source statement

```
REF    ALPH
```

[†]Although a discussion of the object language is not directly pertinent to CP-V, it is included in this manual because it applies to some of the processors operating under CP-V.

it outputs the symbol ALPH, in symbolic (EBCDIC) form, in a declaration specifying that the symbol is an external reference. At this time, the assembler also assigns a declaration name number to the symbol ALPH but does not output the number. The symbol and name number are retained in the assembler's symbol table.

After a symbol has been declared an external reference, it may appear any number of times in the symbolic subprogram in which it was declared. Thus, the use of the symbol ALPH in the source statement

```
    LI,3     ALPH
```

in the above example, is valid even though ALPH is not defined in the subprogram in which it is referenced.

The relocating loader is able to generate interprogram linkages for any symbol that is declared an external definition in the subprogram in which that symbol is defined. Shown below is an example of an external definition in a Symbol source program.

```
            DEF      ALPH
             :
            LI,3     ALPH
             :
    ALPH    AI,4     X'F2'
             :
```

When the assembler processes the source statement

```
    DEF     ALPH
```

it outputs the symbol ALPH, in symbolic (EBCDIC) form, in a declaration specifying that the symbol is an external definition. At this time, the assembler also assigns a declaration name number to the symbol ALPH but does not output the number. The symbol and name number are retained in the assembler's symbol table.

After a symbol has been declared an external definition it may be used (in the subprogram in which it was declared) in the same way as any other symbol. Thus, if ALPH is used as a forward reference, as in the source statement

```
    LI,3      ALPH
```

above, the assembler assigns a forward reference number to ALPH, in addition to the declaration name number assigned previously. (A symbol may be both a forward reference and an external definition.)

On processing the source statement

```
    ALPH     AI,4     X'F2'
```

the assembler outputs the declaration name number of the label ALPH (and an expression for its value) and also outputs the machine-language code for AI,4 and the constant X'F2'.

OBJECT LANGUAGE FORMAT

An object language program generated by a processor is output as a string of bytes representing "load items". A load item consists of an item type code followed by the specific load information pertaining to that item. (The detailed format of each type of load item is given later in this appendix.) The individual load items require varying numbers of bytes for their representation, depending on the type and specific content of each item. A group of 108 bytes, or fewer, comprises a logical record. A load item may be continued from one logical record to the next.

The ordered set of logical records that a processor generates for a program or subprogram is termed an "object module". The end of an object module is indicated by a module-end type code followed by the error severity level assigned to the module by the processor.

RECORD CONTROL INFORMATION

Each record of an object module consists of 4 bytes of control information followed by a maximum of 104 bytes of load information. That is, each record, with the possible exception of the end record, normally consists of 108 bytes of information (i.e., 72 card columns).

The four bytes of control information for each record have the form and sequence shown below.

Byte 0

Record Type			Mode		Format		
			1	1	1	0	0
0	1	2	3	4	5	6	7

Byte 1

Sequence Number

0 7

Byte 2

Checksum

0 7

Byte 3

Record Size

0 7

Record Type specifies whether this record is the last record of the module:

000 means last
001 means not last

Mode specifies that the loader is to read binary information. This code is always 11.

Format specifies object language format. This code is always 100.

Sequence Number is 0 for the first record of the module and is incremented by 1 for each record thereafter, until it recycles to 0 after reaching 255.

Checksum is the computed sum of the bytes comprising the record. Carries out of the most significant bit position of the sum are ignored.

Record Size is the number of bytes (including the record control bytes) comprising the logical record (5 ≤ record size ≤ 108). The record size will normally be 108 bytes for all records except the last one, which may be fewer. Any excess bytes in a physical record are ignored.

LOAD ITEMS

Each load item begins with a control byte that indicates the item type. In some instances, certain parameters are also provided in the load item control byte. In the following discussion, load items are categorized according to their function:

1. Declarations identify to the loader the external and control section labels that are to be defined in the object module being loaded.
2. Definitions define the value of forward references, external definitions, the origin of the subprogram being loaded, and the starting address (e.g., as provided in a Symbol/Meta-Symbol END directive).
3. Expression evaluation load items within a definition provide the values (such as constants, forward references, etc.) that are to be combined to form the final value of the definition.
4. Loading items cause specified information to be stored into core memory.
5. Miscellaneous items comprise padding bytes and the module-end indicator.

DECLARATIONS

In order for the loader to provide the linkage between subprograms, the processor must generate for each external reference or definition a load item, referred to as a "declaration", containing the EBCDIC code representation of the symbol and the information that the symbol is either an external reference or a definition (thus, the loader will have access to the actual symbolic name).

Forward references are always internal references within an object module. (External references are never considered forward references.) The processor does not generate a declaration for a forward reference as it does for externals; however, it does assign name numbers to the symbols referenced.

Declaration name numbers (for control sections and external labels) and forward reference name numbers apply only within the object module in which they are assigned. They have no significance in establishing interprogram linkages, since external references and definitions are correlated by matching symbolic names. Hence, name numbers used in any expressions in a given object module always refer to symbols that have been declared within that module.

The processor must generate a declaration for each symbol that identifies a program section. Each object module produced by an assembler is considered to consist of at least one control section. If no section is explicitly identified in the source program, the assembler assumes it to be a standard control section (discussed below). The standard control section is always assigned a declaration name number of 0. All other control sections (i.e., produced by a processor capable of declaring other control sections) are assigned declaration name numbers (1, 2, 3, etc.) in the order of their appearance in the source program.

In the load items discussed below, the access code, pp, designates the memory protection class that is to be associated with the control section. The meaning of this code is given below.

pp	Memory Protection Feature†
00	Read, write, or access instructions from.
01	Read or access instructions from.
10	Read only.
11	No access.

Control sections are always allocated on a doubleword boundary. The size specification designates the number of bytes to be allocated for the section.

Declare Standard Control Section

Byte 0

Control byte							
0	0	0	0	1	0	1	1
0	1	2	3	4	5	6	7

Byte 1

Access code				Size (bits 1 through 4)			
p	p	0	0				
0	1	2	3	4	5	6	7

Byte 2

Size (bits 5 through 12)	
0	7

Byte 3

Size (bits 13 through 20)	
0	7

This item declares the standard control section for the object module. There may be no more than one standard control section in each object module. The origin of the standard control section is effectively defined when the first reference to the standard control section occurs, although the declaration item might not occur until much later in the object module.

---

† "Read" means a program can obtain information from the protected area; "write" means a program can store information into a protected area; and, "access" means the computer can execute instructions stored in the protected area.

This capability is required by one-pass processors, since the size of a section cannot be determined until all of the load information for that section has been generated by the processor.

Declare Nonstandard Control Section

Byte 0

Control byte							
0	0	0	0	1	1	0	0
0	1	2	3	4	5	6	7

Byte 1

Access code				Size (bits 1 through 4)			
P	P	0	0				
0	1	2	3	4			7

Byte 2

Size (bits 5 through 12)	
0	7

Byte 3

Size (bits 13 through 20)	
0	7

This item declares a control section other than standard control section (see above).

Declare Page Boundary Control Section

Byte 0

Control Byte							
0	0	0	1	1	1	1	0
0	1	2	3	4	5	6	7

Byte 1

Access code				Size (bits 1 through 4)			
P	P	0	0				
0	1	2	3	4	5	6	7

Byte 2

Size (bits 5 through 12)	
0	7

Byte 3

Size (bits 13 through 20)	
0	7

This item declares a nonstandard control section beginning on a memory page boundary.

Declare Dummy Section

Byte 0

Control byte							
0	0	0	0	1	0	0	1
0	1	2	3	4	5	6	7

Byte 1

First byte of name number	
0	7

Byte 2

Second byte of name number†	
0	7

Byte 3

Access code				Size (bits 1 through 4)			
P	P	0	0				
0	1	2	3	4			7

Byte 4

Size (bits 5 through 12)	
0	7

Byte 5

Size (bits 13 through 20)	
0	7

This item comprises a declaration for a dummy control section. It results in the allocation of the specified dummy section, if that section has not been allocated previously by another object module. The label that is to be associated with the first location of the allocated section must be a previously declared external definition name. (Even though the source program may not be required to explicitly designate the label as an external definition, the processor must generate an external definition name declaration for that label prior to generating this load item.)

Declare External Definition Name

Byte 0

Control byte							
0	0	0	0	0	0	1	1
0	1	2	3	4	5	6	7

Byte 1

Name length, in bytes (K)	
0	7

†If the module has fewer than 256 previously assigned name numbers, this byte is absent.

Byte 2

First byte of name (bits 0–7)

⋮

Byte K+1

Last byte of name (bits 0–7)

This item declares a label (in EBCDIC code) that is an external definition within the current object module. The name may not exceed 63 bytes in length.

Declare Primary External Reference Name

Byte 0

Control byte

0	1	2	3	4	5	6	7
0	0	0	0	0	1	0	1

Byte 1

Name length (K), in bytes (bits 0–7)

Byte 2

First byte of name (bits 0–7)

⋮

Byte K+1

Last byte of name (bits 0–7)

This item declares a symbol (in EBCDIC code) that is a primary external reference within the current object module. The name may not exceed 63 bytes in length.

A primary external reference is capable of causing the loader to search the system library for a corresponding external definition. If a corresponding external definition is not found in another load module of the program or in the system library, a load error message is output and the job is errored.

Declare Secondary External Reference Name

Byte 0

Control byte

0	1	2	3	4	5	6	7
0	0	0	0	0	1	1	0

Byte 1

Name length, in bytes (K) (bits 0–7)

Byte 2

First byte of name (bits 0–7)

⋮

Byte K+1

Last byte of name (bits 0–7)

This item declares a symbol (in EBCDIC code) that is a secondary external reference within the current object module. The name may not exceed 63 bytes in length.

A secondary external reference is not capable of causing the loader to search the system library for a corresponding external definition. If a corresponding external definition is not found in another load module of the program, the job is not errored and no error or abnormal message is output.

Secondary external references often appear in library routines that contain optional or alternative subroutines, some of which may not be required by the user's program. By the use of primary external references in the user's program, the user can specify that only those subroutines that are actually required by the current job are to be loaded. Although secondary external references do not cause loading from the library, they do cause linkages to be made between routines that are loaded.

DEFINITIONS

When a source language symbol is to be defined (i.e., equated with a value), the processor provides for such a value by generating an object language expression to be evaluated by the loader. Expressions are of variable length, and terminate with an expression-end control byte (see "Expression Evaluation" in this appendix). An expression is evaluated by the addition or subtraction of values specified by the expression.

Since the loader must derive values for the origin and starting address of a program, these also require definition.

Origin

Byte 0

Control byte

0	1	2	3	4	5	6	7
0	0	0	0	0	1	0	0

This item sets the loader's load-location counter to the value designated by the expression immediately following the origin control byte. This expression must not contain any elements that cannot be evaluated by the loader (see "Expression Evaluation" which follows).

Forward Reference Definition

Byte 0

Control byte

0	1	2	3	4	5	6	7
0	0	0	0	1	0	0	0

Byte 1

First byte of reference number

0 7

Byte 2

Second byte of reference number

0 7

This item defines the value (expression) for a forward reference. The referenced expression is the one immediately following byte 2 of this load item, and must not contain any elements that cannot be evaluated by the loader (see "Expression Evaluation" which follows).

Forward Reference Definition and Hold

Byte 0

Control byte							
0	0	0	1	0	0	0	0
0	1	2	3	4	5	6	7

Byte 1

First byte of reference number

0 7

Byte 2

Second byte of reference number

0 7

This item defines the value (expression) for a forward reference and notifies the loader that this value is to be retained in the loader's symbol table until the module end is encountered. The referenced expression is the one immediately following the name number. It may contain values that have not been defined previously, but all such values must be available to the loader prior to the module end.

After generating this load item, the processor need not retain the value for the forward reference, since that responsibility is then assumed by the loader. However, the processor must retain the symbolic name and forward reference number assigned to the forward reference (until module end).

External Definition

Byte 0

Control byte							
0	0	0	0	1	0	1	0
0	1	2	3	4	5	6	7

Byte 1

First byte of name number

0 7

Byte 2

Second byte of name number†

0 7

This item defines the value (expression) for an external definition name. The name number refers to a previously declared definition name. The referenced expression is the one immediately following the name number.

Define Start

Byte 0

Control byte							
0	0	0	0	1	1	0	1
0	1	2	3	4	5	6	7

This item defines the starting address (expression) to be used at the completion of loading. The referenced expression is the one immediately following the control byte.

EXPRESSION EVALUATION

A processor must generate an object language expression whenever it needs to communicate to the loader one of the following:

1. A program load origin.
2. A program starting address.
3. An external definition value.
4. A forward reference value.
5. A field definition value.

Such expressions may include sums and differences of constants, addresses, and external or forward reference values that, when defined, will themselves be constants or addresses.

After initiation of the expression mode, by the use of a control byte designating one of the five items described above, the value of an expression is expressed as follows:

1. An address value is represented by an offset from the control section base plus the value of the control section base.

†If the module has fewer than 256 previously assigned name numbers, this byte is absent.

2. The value of a constant is added to the accumulated sum by generating an Add Constant (see below) control byte followed by the value, right-justified in four bytes.

   The offset from the control section base is given as a constant representing the number of units of displacement from the control section base, at the resolution of the address of the item. That is, a word address would have its constant portion expressed as a count of the number of words offset from the base, while the constant portion of a byte address would be expressed as the number of bytes offset from the base.

   The control section base value is accumulated by means of an Add Value of Declaration (see below) or Subtract Value of Declaration load item specifying the desired resolution and the declaration number of the control section base. The loader adjusts the base value to the specified address resolution before adding it to the current partial sum for the expression.

   In the case of an absolute address, an Add Absolute Section (see below) or Subtract Absolute Section control byte must be included in the expression to identify the value as an address and to specify its resolution.

3. An external definition of forward reference value is included in an expression by means of a load item adding or subtracting the appropriate declaration or forward reference value. If the value is an address, the resolution specified in the control byte is used to align the value before adding it to the current partial sum for the expression. If the value is a constant, no alignment is necessary.

Expressions are not evaluated by the loader until all required values are available. In evaluating an expression, the loader maintains a count of the number of values added or subtracted at each of the four possible resolutions. A separate counter is used for each resolution, and each counter is incremented or decremented by 1 whenever a value of the corresponding resolution is added to or subtracted from the loader's expression accumulator. The final accumulated sum is a constant, rather than an address value, if the final count in all four counters is equal to 0. If the final count in one (and only one) of the four counters is equal to +1 or -1, the accumulated sum is a "simple address" having the resolution of the nonzero counter. If more than one of the four counters hava a nonzero final count, the accumulated sum is termed a "mixed-resolution expression" and is treated as a constant rather than an address.

The resolution of a simple address may be altered by means of a Change Expression Resolution (see below) control byte. However, if the current partial sum is either a constant or a mixed-resolution value when the Change Expression Resolution control byte occurs, then the expression resolution is unaffected.

Note that the expression for a program load origin or starting address must resolve to a simple address, and the single nonzero resolution counter must have a final count of +1 when such expressions are evaluated.

In converting a byte address to a word address, the two least significant bits of the address are truncated. Thus, if the resulting word address is later changed back to byte resolution, the referenced byte location will then be the first byte (byte 0) of the word.

After an expression has been evaluated, its final value is associated with the appropriate load item.

In the following diagrams of load item formats, RR refers to the address resolution code. The meaning of this code is given in the table below.

RR	Address Resolution
00	Byte
01	Halfword
10	Word
11	Doubleword

The load item discussed in this appendix, "Expression Evaluation", may appear only in expressions.

Add Constant

Byte 0

Control byte							
0	0	0	0	0	0	0	1
0	1	2	3	4	5	6	7

Byte 1

First byte of constant	
0	7

Byte 2

Second byte of constant	
0	7

Byte 3

Third byte of constant	
0	7

Byte 4

Fourth byte of constant	
0	7

This item causes the specified four-byte constant to be added to the loader's expression accumulator. Negative constants are represented in two's complement form.

Add Absolute Section

Byte 0

Control byte							
0	0	1	1	0	1	R	R
0	1	2	3	4	5	6	7

This item identifies the associated value (expression) as a positive absolute address. The address resolution code, RR, designates the desired resolution.

Subtract Absolute Section

Byte 0

Control byte							
0	0	1	1	1	0	R	R
0	1	2	3	4	5	6	7

This item identifies the associated value (expression) as a negative absolute address. The address resolution code, RR, designates the desired resolution.

Add Value of Declaration

Byte 0

Control byte							
0	0	1	0	0	0	R	R
0	1	2	3	4	5	6	7

Byte 1

First byte of name number	
0	7

Byte 2

Second byte of name number†	
0	7

†If the module has fewer than 256 previously assigned name numbers, this byte is absent.

This item causes the value of the specified declaration to be added to the loader's expression accumulator. The address resolution code, RR, designates the desired resolution, and the name number refers to a previously declared definition name that is to be associated with the first location of the allocated section.

One such item must appear in each expression for a relocatable address occurring within a control section, adding the value of the specified control section declaration (i.e., adding the byte address of the first location of the control section).

Add Value of Forward Reference

Byte 0

Control byte							
0	0	1	0	0	1	R	R
0	1	2	3	4	5	6	7

Byte 1

First byte of forward reference number	
0	7

Byte 2

Second byte of forward reference number	
0	7

This item causes the value of the specified forward reference to be added to the loader's expression accumulator. The address resolution code, RR, designates the desired resolution, and the designated forward reference must not have been defined previously.

Subtract Value of Declaration

Byte 0

Control byte							
0	0	1	0	1	0	R	R
0	1	2	3	4	5	6	7

Byte 1

First byte of name number	
0	7

Byte 2

Second byte of name number†

This item causes the value of the specified declaration to be subtracted from the loader's expression accumulator. The address resolution code, RR, designates the desired resolution, and the name number refers to a previously declared definition name that is to be associated with the first location of the allocated section.

Subtract Value of Forward Reference

Byte 0

Control byte							
0	0	1	0	1	1	R	R
0	1	2	3	4	5	6	7

Byte 1

First byte of forward reference number	
0	7

Byte 2

Second byte of forward reference number	
0	7

This item causes the value of the specified forward reference to be subtracted from the loader's expression accumulator. The address resolution code, RR, designates the desired resolution, and the designated forward reference must not have been defined previously.

Change Expression Resolution

Byte 0

Control byte							
0	0	1	1	0	0	R	R
0	1	2	3	4	5	6	7

This item causes the address resolution in the expression to be changed to that designated by RR.

Expression End

Byte 0

Control byte							
0	0	0	0	0	0	1	0
0	1	2	3	4	5	6	7

This item identifies the end of an expression (the value of which is contained in the loader's expression accumulator).

FORMATION OF INTERNAL SYMBOL TABLES

The three object code control bytes described below are required to supply the information necessary in the formation of Internal Symbol Tables.

In the following diagrams of load item formats, Type refers to the symbol types supplied by the object language and maintained in the symbol table. IR refers to the internal resolution code. Type and resolution are meaningful only when the value of a symbol is an address. In this case, it is highly likely that the processor knows the type of value that is in the associated memory location, and the type field identifies it. The resolution field indicates the resolution of the location counter at the time the symbol was defined. The following tables summarize the combinations of value and meaning.

Symbol Types

Type	Meaning of 5-Bit Code
00000	Instruction
00001	Integer
00010	Short floating point
00011	Long floating point
00110	Hexadecimal (also for packed decimal)
00111	EBCDIC text (also for unpacked decimal)
01001	Integer array
01010	Short floating-point array
01011	Long floating-complex array
01000	Logical array
10000	Undefined symbol

Internal Resolution

IR	Address Resolution
000	Byte
001	Halfword
010	Word
011	Doubleword
100	Constant

Type Information for External Symbol

Byte 0

Control byte							
0	0	0	1	0	0	0	1
0	1	2	3	4	5	6	7

Byte 1

Type field		IR field	
0	4	5	7

Byte 2

Name number	
0	7

Byte 3 (if required)

Name number (continued)	
0	7

This item provides type information for external symbols. The Type and IR fields are defined above. The name number field consists of one or two bytes (depending on the current declaration count) which specifies the declaration number of the external definition.

Type and EBCDIC for Internal Symbol

Byte 0

Control byte							
0	0	0	1	0	0	1	0
0	1	2	3	4	5	6	7

Byte 1

Type field	IR field
0 4	5 7

Byte 2

Length of name (EBCDIC characters)
0 7

Byte 3

First byte of name in EBCDIC
0 7

Byte n

Last byte of name in EBCDIC
0 7

Byte n + 1, . . .

Expression defining value of internal symbol
0 7

This item supplies type and EBCDIC for an internal symbol. The load items for Type and IR are as above. Length of name specifies the length of the EBCDIC name in characters. The name, in EBCDIC, is specified in the required number of bytes, followed by the expression defining the internal symbol.

EBCDIC for an Undefined Symbol

Byte 0

Control byte							
0	0	0	1	0	0	1	1
0	1	2	3	4	5	6	7

Byte 1

Length of name (EBCDIC characters)
0 7

Byte 2

First byte of name in EBCDIC
0 7

Byte n

Last byte of name in EBCDIC
0 7

Byte n + 1, n + 2

Two bytes of symbol associated forward reference number
0 7

This item is used to associate a symbol with a forward reference. The length of name and name in EBCDIC are the same as in the above item. The last two bytes specify the forward reference number with which the above symbol is to be associated.

LOADING

Load Absolute

Byte 0

Control byte							
0	1	0	0	N	N	N	N
0	1	2	3	4	5	6	7

Byte 1

First byte to be loaded
0 7

⋮

Byte NNNN

Last byte to be loaded
0 7

This item causes the next NNNN bytes to be loaded absolutely (NNNN is expressed in natural binary form, except that 0000 is interpreted as 16 rather than 0). The load location counter is advanced appropriately.

Load Relocatable (Long Form)

Byte 0

Control byte							
0	1	0	1	Q	C	R	R
0	1	2	3	4	5	6	7

Byte 1

First byte of name number
0 7

Byte 2

Second byte of name number†
0 7

This item causes a four-byte word (immediately following this load item) to be loaded, and relocates the address field according to the address resolution code, RR. Control bit C designates whether relocation is to be relative to a forward reference (C = 1) or relative to a declaration (C = 0). Control bit Q designates whether a 1-byte (Q = 1) or a 2-byte (Q = 0) name number follows the control byte of this load item.

†If the module has fewer than 256 previously assigned name numbers, this byte is absent.

If relocation is to be relative to a forward reference, the forward reference must not have been defined previously. When this load item is encountered by the loader, the load location counter can be aligned with a word boundary by loading the appropriate number of bytes containing all zeros (e.g., by means of a load absolute item).

Load Relocatable (Short Form)

Byte 0

Control byte							
1	C	D	D	D	D	D	D
0	1	2	3	4	5	6	7

This item causes a four-byte word (immediately following this load item) to be loaded, and relocates the address field (word resolution). Control bit C designates whether relocation is to be relative to a forward reference (C = 1) or relative to a declaration (C = 0). The binary number DDDDDD is the forward reference number or declaration number by which relocation is to be accomplished.

If relocation is to be relative to a forward reference, the forward reference must not have been defined previously. When this load item is encountered by the loader, the load location counter must be on a word boundary (see "Load Relocatable (Long Form)", above).

Repeat Load

Byte 0

Control byte							
0	0	0	0	1	1	1	1
0	1	2	3	4	5	6	7

Byte 1

First byte of repeat count	
0	7

Byte 2

Second byte of repeat count	
0	7

This item causes the loader to repeat (i.e., perform) the subsequent load item a specified number of times. The repeat count must be greater than 0, and the load item to be repeated must follow the repeat load item immediately.

Define Field

Byte 0

Control byte							
0	0	0	0	0	1	1	1
0	1	2	3	4	5	6	7

Byte 1

Field location constant, in bits (K)	
0	7

Byte 2

Field length, in bits (L)	
0	7

This item defines a value (expression) to be added to a field in previously loaded information. The field is of length L ($1 \leq L \leq 255$) and terminates in bit position T, where:

T = current load bit position −256 +K.

The field location constant, K, may have any value from 1 to 255. The expression to be added to the specified field is the one immediately following byte 2 of this load item.

MISCELLANEOUS LOAD ITEMS

Padding

Byte 0

Control byte							
0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7

Padding bytes are ignored by the loader. The object language allows padding as a convenience for processors.

Module End

Byte 0

Control byte							
0	0	0	0	1	1	1	0
0	1	2	3	4	5	6	7

Byte 1

Severity level							
0	0	0	0	E	E	E	E
0	1	2	3	4	5	6	7

This item identifies the end of the object module. The value EEEE is the error severity level assigned to the module by the processor.

OBJECT MODULE EXAMPLE

The following example shows the correspondence between the statements of a Meta-Symbol source program and the string of object bytes output for that program by the assembler. The program, listed below, has no significance other than illustrating typical object code sequences.

Example

1					DEF	AA, BB, CC	CC IS UNDEFINED BUT CAUSES NO ERROR
2					REF	RZ, RTN	EXTERNAL REFERENCES DECLARED
3	00000			ALPHA	CSECT		DEFINE CONTROL SECTION ALPHA
4	000C8				ORG	200	DEFINE ORGIN
5	000C8	22000000	N	AA	LI, CNT	0	DEFINES EXTERNAL AA; CNT IS A FWD REF
6	000C9	32000000	N		LW, R	RZ	R IS A FORWARD REFERENCE;
7				*			RZ IS AN EXTERNAL REFERENCE, AS
8				*			DECLARED IN LINE 2
9	000CA	50000000	N	RPT	AH, R	KON	DEFINES RPT; R AND KON ARE
10				*			FORWARD REFERENCES
11	000CB	69200000	F		BCS, 2	BB	BB IS AN EXTERNAL DEFINITION
12				*			USED AS A FORWARD REFERENCE
13	000CC	20000001	N		AI, CNT	1	CNT IS A FORWARD REFERENCE
14	000CD	680000CA			B	RPT	RPT IS A BACKWARD REFERENCE
15	000CE	68000000	X		B	RTN	RTN IS AN EXTERNAL REFERENCE
16	000CF	0001	A	KON	DATA, 2	1	DEFINES KON
17		00000003		R	EQU	3	DEFINES R
18		00000004		CNT	EQU	4	DEFINES CNT
19	000D0	224FFFFF	A	BB	LI, CNT	-1	DEFINES EXTERNAL BB THAT HAS
20				*			ALSO BEEN USED AS A FORWARD
21				*			REFERENCE
22	000C8				END	AA	END OF PROGRAM

CONTROL BYTES (In Binary)

Begin Record	Record number: 0		
00111100	Record type: not last, Mode binary, Format: object language.		Record control information not part of load item
00000000	Sequence number 0		
01100011	Checksum: 99		
01101100	Record size: 108		
	0302C1C1 (hexadecimal code comprising the load item)		Source Line 1
00000011	Declare external definition name (2 bytes) Name: AA	Declaration number: 1	
	0302C2C2		
00000011	Declare external definition name (2 bytes) Name: BB	Declaration number: 2	
	0302C3C3		
00000011	Declare external definition name (2 bytes) Name: CC	Declaration number: 3	
	0502D9E9		Source Line 2
00000101	Declare primary reference name (2 bytes) Name RZ	Declaration number: 4	
	0503D9E3D5		
00000101	Declare primary reference name (3 bytes) Name: RTN	Declaration number: 5	

Begin Record	Record number: 0	
	0A010100000320200002	Source Line 5[†]
00001010	Define external definition	
	Number 1	
00000001	Add constant: 800 X'320'	
00100000	Add value of declaration (byte resolution)	
	Number 0	
00000010	Expression end	
	040100000320200002	Source Line 4
00000100	Origin	
00000001	Add constant: 800 X'320'	
00100000	Add value of declaration (byte resolution)	
	Number 0	
00000010	Expression end	
	4422000000	Source Line 5
01000100	Load absolute the following 4 bytes: X'22000000'	
	07EB0426000002	
00000111	Define field	
	Field location constant: 235 bits	
	Field length: 4 bits	
	Add the following expression to the above field:	
00100110	Add value of forward reference (word resolution)	
	Number 0	
00000010	Expression end	
	8432000000	Source Line 6
10000100	Load relocatable (short form). Relocate address field (word resolution)	
	Relative to declaration number 4	
	The following 4 bytes: X'32000000'	
	07EB0426000602	
00000111	Define field	
	Field location constant: 235 bits	
	Field length: 4 bits	
	Add the following expression to the above field:	
00100110	Add value of forward reference (word resolution)	
	Number 6	
00000010	Expression end	
	CC50000000	Source Line 9
11001100	Load relocatable (short form). Relocate address field (word resolution)	
	Relative to forward reference number 12	
	The following 4 bytes: X'50000000'	
	07EB0426000602	
00000111	Define field	
	Field location constant: 235 bits	
	Field length: 4 bits	
	Add the following expression to the above field:	
00100110	Add value of forward reference (word resolution)	
	Number 6	
00000010	Expression end	

[†] No object code is generated for source lines 3 (define control section) or 4 (define origin) at the time they are encountered. The control section is declared at the end of the program after Symbol has determined the number of bytes the program requires. The origin definition is generated prior to the first instruction.

Begin Record	Record number: 0	
	D269200000	Source Line 11
11010010	Load relocatable (short form). **Relocate address field (word resolution)**	
	Relative to forward reference number 18	
	The following 4 bytes: X'69200000'	
	4420000001	Source Line 13
01000100	Load absolute the following 4 bytes: X'20000001'	
	07EB0426000002	
00000111	Define field	
	Field location constant: 235 bits	
	Field length: 4 bits	
	Add the following expression to the above field:	
00100110	Add value of forward reference (word resolution)	
	Number 0	
00000010	Expression end	
	80680000CA	Source Line 14
10000000	Load relocatable (short form). Relocate address field (word resolution)	
	Relative to declaration number 0	
	The following 4 bytes: X'680000CA'	
	8568000000	Source Line 15
10000101	Load relocatable (short form). Relocate address field (word resolution)	
	Relative to declaration number 5	
	The following 4 bytes: X'68000000'	
	08	Source Line 16
00001000	Define forward reference (continued in record 1)	

Begin Record	Record number: 1	
00011100	Record type: last, Mode: binary, Format: object language.	Record Control Information
00000001	Sequence number 1	
11101100	Checksum: 236	
01010001	Record size: 81	
	000C010000033C200002 (continued from record 0)	Source Line 16
	Number 12	
00000001	Add constant: 828 X'33C'	
00100000	Add value of declaration (byte resolution)	
	Number 0	
00000010	Expression end	
	42001	
01000010	Load absolute the following 2 bytes: X'0001'	
	080006010000000302	Source Line 17
00001000	Define forward reference	
	Number 6	
00000001	Add constant: 3 X'3'	
00000010	Expression end	
	080000010000000402	Source Line 18
00001000	Define forward reference	
	Number 0	
00000001	Add constant: 4 X'4'	
00000010	Expression end	

Begin Record	Record number: 1	
	0F00024100	Advance to Word Boundary
00001111	Repeat load	
	Repeat count: 2	
01000001	Load absolute the following 1 bytes: X'00'	
	080012010000340200002	Source Line 19
00001000	Define forward reference	
	Number 18	
00000001	Add constant: 832 X'340'	
	Add value of declaration (byte resolution)	
	Number 0	
00000010	Expression end	
	0A020100000340200002	
00001010	Define external definition	
	Number 2	
00000001	Add constant: 832 X'340'	
00100000	Add value of declaration (byte resolution)	
	Number 0	
00000010	Expression end	
	44224FFFFF	
01000100	Load absolute the following 4 bytes: X'224FFFFF'	
	0D0100000320200002	Source Line 22
00001101	Define start	
00000001	Add constant: 800 X'320'	
00100000	Add value of declaration (byte resolution)	
	Number 0	
00000010	Expression end	
	0B000344	
00001011	Declare standard control section declaration number: 0	
	Access code: Full access. Size 836 X'344'	
	0E00	
00001110	Module end	
	Severity level: X'0'	

A table summarizing control byte codes for object language load items is given below.

Object Code Control Byte	Type of Load Item
0 0 0 0 0 0 0 0	Padding
0 0 0 0 0 0 0 1	Add constant
0 0 0 0 0 0 1 0	Expression end
0 0 0 0 0 0 1 1	Declare external definition name
0 0 0 0 0 1 0 0	Origin
0 0 0 0 0 1 0 1	Declare primary reference name
0 0 0 0 0 1 1 0	Declare secondary reference name
0 0 0 0 0 1 1 1	Define field
0 0 0 0 1 0 0 0	Define forward reference
0 0 0 0 1 0 0 1	Declare dummy section
0 0 0 0 1 0 1 0	Define external definition

Object Code Control Byte								Type of Load Item
0	0	0	0	1	0	1	1	Declare standard control section
0	0	0	0	1	1	0	0	Declare nonstandard control section
0	0	0	0	1	1	0	1	Define start
0	0	0	0	1	1	1	0	Module end
0	0	0	0	1	1	1	1	Repeat load
0	0	0	1	0	0	0	0	Define forward reference and hold
0	0	0	1	0	0	0	1	Provide type information for external symbol
0	0	0	1	0	0	1	0	Provide type and EBCDIC for internal symbol
0	0	0	1	0	0	1	1	EBCDIC and forward reference number for undefined symbol
0	0	0	1	1	1	1	0	Declare page boundary control section
0	0	1	0	0	0	R	R	Add value of declaration
0	0	1	0	0	1	R	R	Add value of forward reference
0	0	1	0	1	0	R	R	Subtract value of declaration
0	0	1	0	1	1	R	R	Subtract value of forward reference
0	0	1	1	0	0	R	R	Change expression resolution
0	0	1	1	0	1	R	R	Add absolute section
0	0	1	1	1	0	R	R	Subtract absolute section
0	1	0	0	N	N	N	N	Load absolute
0	1	0	1	Q	C	R	R	Load relocatable (long form)
1	C	D	D	D	D	D	D	Load relocatable (short form)

APPENDIX E. SIGMA STANDARD COMPRESSED LANGUAGE

The Sigma Standard Compressed Language is used to represent source EBCDIC information in a highly compressed form.

Meta-Symbol (along with several of the utility programs) accepts this form as input or output, will accept updates to the compressed input and will regenerate source when requested. No information is destroyed in the compression or decompression.

Records may not exceed 108 bytes in length. Compressed records are punched in the binary mode when represented on card media. Therefore, on cards, columns 73 through 80 are not used and are available for comment or identification information.

The first four bytes of each record are for checking purposes. They are as follows:

Byte 1 Identification (00L11000) L = 1 for each record except the last record, in which case L = 0.

Byte 2 Sequence number (0 to 255 and recycles).

Byte 3 Checksum which is the least significant 8 bits of the sum of all bytes in the record except the checksum byte itself. Carries out of the most significant bit are ignored. If the checksum byte is all 1's, do not checksum the record.

Byte 4 Number of bytes comprising record including the checking bytes ($\leq$ 108)

The rest of the record consists of a string of 6-bit and 8-bit items. Any partial item at the end of a record is ignored.

The following six-bit items (decimal number assigned) comprise the string control:

Item	Function	Item	Function
0	Ignore	32	O
1	Not currently assigned	33	P
2	End of line	34	Q
3	End of file	35	R
4	Use 8-bit character that follows[†]	36	S
5	Use n + 1 blanks (next 6-bit item is n)	37	T
6	Use n + 65 blanks (next 6-bit item is n)	38	U
7	Blank	39	V
8	0	40	W
9	1	41	X
10	2	42	Y
11	3	43	Z
12	4	44	.
13	5	45	<
14	6	46	(
15	7	47	+
16	8	48	|
17	9	49	&
18	A	50	$
19	B	51	*
20	C	52	)
21	D	53	;
22	E	54	¬
23	F	55	-
24	G	56	/
25	H	57	,
26	I	58	%
27	J	59	⌴
28	K	60	>
29	L	61	:
30	M	62	'
31	N	63	=

[†]Eight-bit characters are in uncompressed EBCDIC format (e.g., !@#?).

INDEX

Note: For each entry in this index, the number of the most significant page is listed first. Any pages thereafter are listed in numerical sequence.

Note: For each entry in this index, the number of the most significant page is listed first. Any pages thereafter are listed in numerical sequence.

Note: For each entry in this index, the number of the most significant page is listed first. Any pages thereafter are listed in numerical sequence.

E

Note: For each entry in this index, the number of the most significant page is listed first. Any pages thereafter are listed in numerical sequence.

F

G

H

I

J

K

L

Note: For each entry in this index, the number of the most significant page is listed first. Any pages thereafter are listed in numerical sequence.

M

N

O

P

Note: For each entry in this index, the number of the most significant page is listed first. Any pages thereafter are listed in numerical sequence.

Q

R

S

Note: For each entry in this index, the number of the most significant page is listed first. Any pages thereafter are listed in numerical sequence.

Note: For each entry in this index, the number of the most significant page is listed first. Any pages thereafter are listed in numerical sequence.