Xerox Control Program-Five (CP-V)

Xerox 560 and Sigma 6/7/9 Computers

System Management

Reference Manual

90 16 74H

XEROX

Xerox Control Program-Five (CP-V)

Xerox 560 and Sigma 5/6/7/9 Computers

System Management

Reference Manual

90 16 74H

90 16 74H-1

90 16 74H-2

September 1978

XL21B, Rev. 0
File No.: 1X13
Printed in U S A

REVISION

This publication documents the F00 version of Control Program-Five (CP-V). The publication consists of the H edition of this manual (90 16 74H, dated November 1975) and the revision packages numbered 90 16 74H-1 (11/76) and 90 16 74H-2(9/78). Vertical lines in the margins of pages labeled 90 16 74H-2(9/78) indicate technical changes that reflect the F00 version of CP-V. Vertical lines in the margins of other pages indicate changes that occurred in a previous release of the system.

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)/TP Reference Manual	90 31 12
Xerox Control Program-Five (CP-V)/SP Reference Manual	90 31 13
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 Assembly Program/Reference Manual	90 30 00
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, SP - system programming, TP - transaction processing, TS - time-sharing, UT - utilities.

The specifications of the software system described in this publication are subject to change without notice. The availability or performance of sor features may depend on a specific configuration of equipment such as additional tape units or larger memory. Customers should consult their sale representative for details.

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 generating a CP-V system (SYSGEN), authorizing users, maintaining user accounting records, maintaining the file system, monitoring system performance, and other related functions.

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

- The CP-V Batch Processing 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 System Programming Reference Manual, 90 31 13, describes the CP-V features that are designed to aid the system programmer in the development, maintenance, and modification of the CP-V system.

- The CP-V Transaction Processing Reference Manual, 90 31 12, provides information about dynamically modifying and querying a central database in a transaction processing environment. The manual is addressed to system managers, database administrators, applications programmers, and computer operators.

- 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. {A / id} 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. !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. !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.

language processor — a program that translates user's source program(s) into object language.

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 language 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 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 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 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 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 Xerox 560 and 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, transaction 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 CPUs, multiple I/O processors, and device pooling.
- A complete recovery system coupled with preservation of user files to provide fast restart following hardware or software 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, a complete set of powerful processors, and compatability between on-line and batch modes for all processors and data base (file) handling.

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 or another computer.

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.
AP	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.)
LEMUR	Construction and manipulation of unshared libraries.

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 establishment of communication networks.

TRANSACTION PROCESSING

The transaction processing feature of CP-V is an economical approach to efficient centralized information processing. It is a generalized package that fulfills the requirements of a variety of business applications. Transaction processing facilities provide an environment in which several users at remote terminals may enter business transactions, simultaneously utilizing a common database. The transactions are processed immediately, as they are received, by application programs written especially for the particular installation. As necessary, reports are created and sent to an appropriate remote terminal. (Reference: CP-V/TP Reference Manual, 90 31 12.)

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 a CP-V installation must evaluate the performance requirements before he can effectively use the system management facilities. This evaluation must precede equipment selection since an effective equipment selection can be made only with complete knowledge of the intended use of CP-V.

The performance requirements must define 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 acceptable batch turnaround time and the tolerable interactive delays must be defined. Other information 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 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. Help determine the relative importance of various parts of the system in terms of CPU use 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. Additionally, the system manager may also individually authorize or deny access to the various processors for each user.

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 and defined in the following paragraphs.

COMMAND PROCESSORS

There are four command processors: 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.

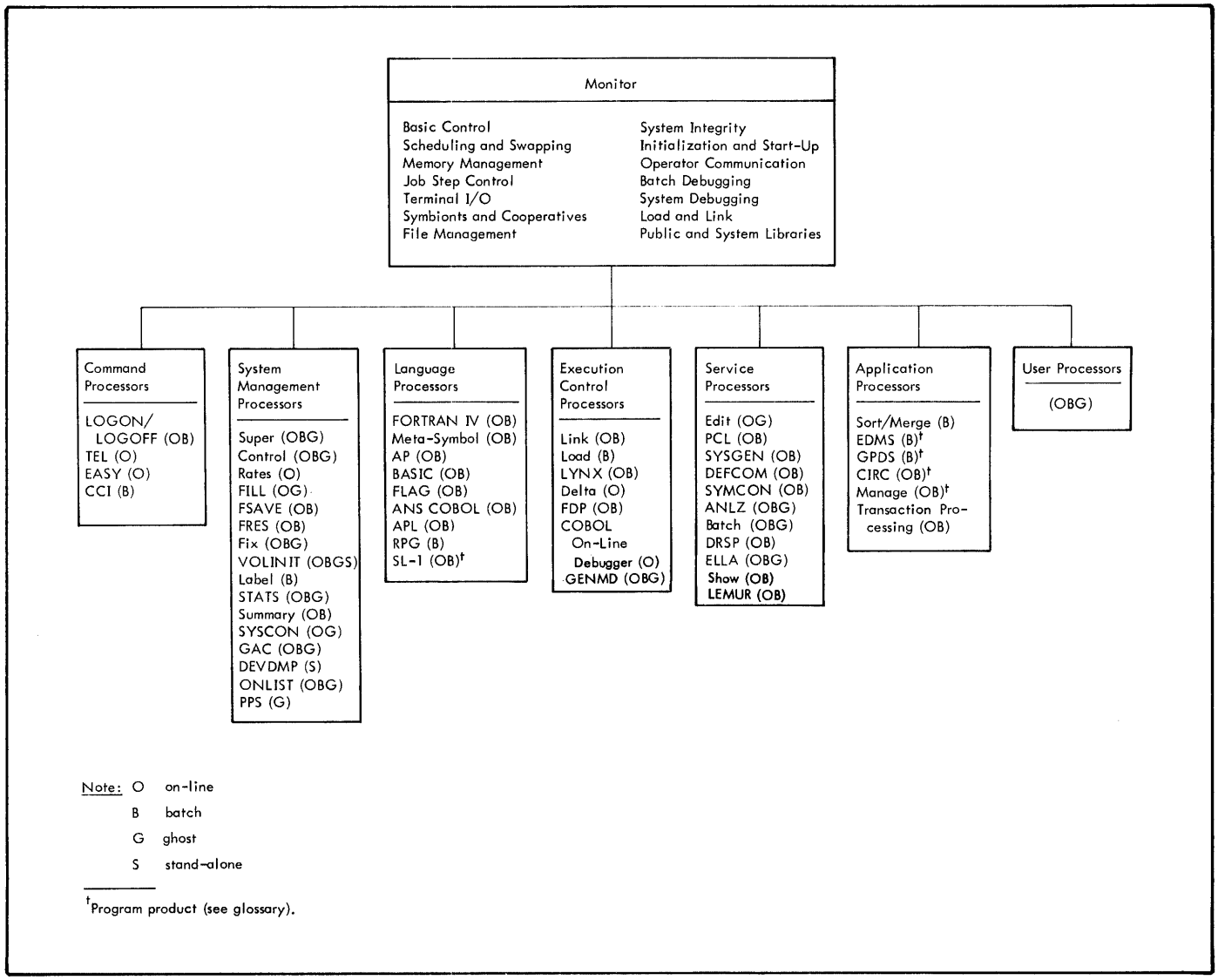


Figure 1. CP-V Operating System

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. Fourteen 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, specify how many central site magnetic tape units a user will have, grant certain users, such as system programmers, special privileges, (e.g., the privilege of examining, accessing, and changing the monitor), and individually authorize or deny access to the various processors for each user. Super is also used to create and delete remote processing workstations. (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 or Fill. (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 dis- ,ed played 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.)

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 7.)

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.)

FIX

The Fix processor enables the system manager to repair or delete damaged file directories. It also provides HGP reconstruction for private disk pack sets and the public file system. (Reference: CP-V/OPS Reference Manual, 90 16 75.)

DEVDMP

The Device Save/Restore processor (DEVDMP) is a standalone 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.)

ONLIST

The ONLIST processor is invoked by a system management account in the batch, ghost, or on-line mode to display the contents of the :LOGD file used for the TEL WHERE command. This file, created and updated by LOGON, contains one record for each on-line user. The records are keyed by the users' sysid. Each record contains the user's line number, name, account, and the time the user logged on or off. Since LOGON accesses the :LOGD file in shared update mode, ONLIST should be used to list the file rather than PCL to avoid delaying LOGON. When invoked on-line, ONLIST displays only those users currently logged on. When invoked in batch or ghost mode, all records in the file (those of both logged on and logged off users) will be listed. In batch mode, records for logged off users are deleted from the file. These listings are produced simply by calling the ONLIST processor. No commands are required.

PHYSICAL PAGE STEALER (PPS)

The Physical Page Stealer is a ghost job which is used for management of all dedicated foreground memory in real-time systems. PPS allows the user to display memory segments currently allocated, get DYNRESDF pages, free DYNRESDF pages, and redefine the RESDF area. (Reference: CP-V/SP Reference Manual, 90 31 13.)

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 produces efficient object code, thus reducing execution time and core requirements, and generates 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 that 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 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.)

AP

Assembly Program (AP) is a four-phase assembler that reads source language programs and converts them to object language programs. AP outputs the object language program, an assembly listing, and a cross reference (or concordance listing. AP is available in both the on-line and batch modes.

The following list summarizes AP's more important features for the programmer:

- Self-defining constants that facilitate use of hexadecimal, decimal, octal, floating-point, scaled fixed-point, and text string values.
- The facility for writing large programs in segments or modules. The assembler will provide information necessary for the loader to complete the linkage between modules when they are loaded into memory.

- The label, command, and argument fields may contain both arithmetic and logical expressions, using constant or variable quantities.

- Full use of lists and subscripted elements is provided.

- The DO, DO1, and GOTO directives allow selective generation of areas of code, with parametric constants or expressions evaluated at assembly time.

- Command procedures allow the capability of generating many units of code for a given procedure call line.

- Function procedures return values to the procedure call line. They also provide the capability of generating many units of code for a given procedure call line.

- Individual parameters on a procedure call line can be tested both arithmetically and logically.

- Procedures may call other procedures, and may call procedures recursively.

(Reference: Xerox Assembly Program Reference Manual, 90 30 00.)

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. Programs are usually small to medium size applications of a computational nature.

BASIC is used 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 rerouted, 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 allows 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 is 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 the machine'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. Libro v 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.)

RPG

Xerox RPG (Report Program Generator) is a convenient means of preparing reports from information available in computer-readable forms, such as punched cards, magnetic tape, and magnetic disks. In addition, it is a means of establishing and updating files of information, usually in conjunction with preparation of reports.

RPG provides its capabilities through generation (compilation) of object programs, each of which is tailored to produce a different set of reporting results and/or file processing desired by the user. The RPG object programs are capable of accepting input data, retrieving data from existing files, performing calculations, changing formats of data, updating existing files, creating new files, comparing data values to one another and to specified constants to determine appropriate handling, using user-defined processing subroutines, using system library subroutines, and printing reports derived from the input and file data.

Xerox RPG has several advantages over the more traditional method of writing object programs in a symbolic programming language. The RPG language is oriented toward the user's problem, describing reporting requirements, rather than toward the mechanics and manipulations of computer usage. The language and specification techniques are easily learned. A user can become proficient in RPG after writing only a few programs, whereas an equal facility in symbolic programming would require considerable experience. (Reference: RPG/ Reference Manual, 90 19 99.)

SIMULATION LANGUAGE (PROGRAM PRODUCT)[†]

The Simulation Language (SL-1) is a simplified, problem-oriented digital programming language designed specifically

[†]See "program product" in glossary.

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 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 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.)

LYNX

LYNX is a load processor that is available in both the on-line and batch modes. LYNX has most of the capabilities of the Load loader and also provides the same control over internal and global symbol table construction which is available in the Link loader. LYNX may be viewed as a preprocessor for the Load loader. After it analyzes the user's commands, it constructs a table of loader control information which it then passes to the Load loader. It is the Load loader which actually performs the loading process. (Reference: CP-V/BP Reference Manual, 90 17 64.)

GENMD

GENMD permits on-line, batch, and ghost users to make permanent modifications to existing load modules, thereby reducing the number of compilations required to debug a program. (Reference: CP-V/SP Reference Manual, 90 31 13.)

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 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 permits insertion, deletion, reordering, and replacement of lines or groups of lines of text. It also permits 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 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 may 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 9.)

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.)

ANLZ

ANLZ provides the system programmer with a means of examining and analyzing 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 crash dumps or the currently running monitor. (Reference: CP-V/SP Reference Manual, 90 31 13.)

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.)

DRSP

DRSP (Dynamic Replacement of Shared Processors) enables the system programmer to dynamically add, replace, or delete processors during normal system operation with other users in the system. (Reference: CP-V/SP Reference Manual 90 31 13.)

ELLA

The Error Log Listing program (ELLA) provides an efficient tool to list and sort the error data base which is automatically generated and updated by the CP-V system. (Reference: CP-V/SP Reference Manual, 90 31 13.)

SHOW

The Show processor allows the user to display his current maximum system services and resources, the peripheral devices that he has been authorized to use, and several other system user parameters. (Reference: CP-V/SP Reference Manual, 90 31 13.)

LEMUR

LEMUR (Library Editor and Maintenance Utility Routine) lets the user construct, copy, or delete library modules of either the ROM or load module type. (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.)

EDMS (PROGRAM PRODUCT)†

EDMS is a generalized data management system that enables the user to create an integrated data base. It may 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.
- 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: 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 and require little or no knowledge of programming for execution.

†See "program product" in glossary.

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.)

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.)

TRANSACTION PROCESSING

Transaction Processing is designed for applications that require the entry and processing of on-line transactions. It is a collection of general-purpose components and supporting monitor services available under the CP-V operating system. Transaction Processing (TP) enables business to move from cyclic batch processing to remote on-line operations, where transactions are entered directly from their point of origin. The Xerox system consists of

- The CP-V monitor and standard processors such as COBOL, Meta-Symbol, and FORTRAN.
- Terminal Interface Controller.
- Utility processors that create files for external system control.
- Transaction Processing Controller.
- Extended Data Management System (EDMS).

(Reference: CP-V/TP Reference Manual, 90 31 12.)

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 CP-V/SP Reference Manual, 90 31 13.

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.

[†]See "program product" in glossary.

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. Checking user authorization for individual processor.

6. 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 two major functions.

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

   b. 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 types of FORTRAN libraries. One is a public library and the other is a system library. In the standard release of CP-V, there are three FORTRAN public libraries. One library (P1) contains a useful set of Extended FORTRAN IV run-time library routines; another (P0) contains P1 and the FORTRAN Debug Package; the third (P4) contains P1 and the FORTRAN real-time features. These three 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.

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

Event	Meaning
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.
EQFAC	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 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 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.

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.

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 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.

The allocation scheme for systems which have file space allocated on both RADs and disk packs is described in the following paragraphs.

For overall performance, the RAD is preferred for frequently accessed system information and temporary files used by major processors. Special users who need high performance on special files may specify RAD preference.

All of the account directory and all files for :SYS are assigned to the RAD. The first granule of each file directory is assigned to disk pack but any additional granules are assigned to RAD. All star or id files and all scratch files (opened OUT or OUTIN with REL) prefer RAD. Random files with no user stated preference and all other files and their indexes prefer pack. These pack preferences may be overridden either by the operator key-in 'PREFER' for all files or by the user specification of NOSEP and DEVICE for individual files.

Briefly, the effect of authorization and defaults upon the allocation is: If not enough space is available on the preferred device, the other device will be used if space is available. The exception to this is random files with user specified preference. In this case, if space is not available on the user specified device, the file is not allocated and an error is returned to the user. Also, within the authorized limits, temporary files may use only temporary authorization and permanent files may use only permanent authorization.

The general rule for authorization should be: A large amount of temporary RAD and disk pack space should be authorized for all users and the amount of permanent disk space should be individually authorized by need. Very few users should be authorized permanent RAD space.

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 address mapping hardware, access protection, and write locks in allocating available physical core pages to users. 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.

SYSTEM INTEGRITY

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

8. For the Xerox 560 – on-line interface for remote assistance.

ERROR DETECTION AND RECOVERY

An effective 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. Some hardware errors, such as loss of a memory power supply, lead to system shutdown.

Software consistency checks, some of which are performed optionally on the setting 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).

The error messages generated throughout the system (reporting both hardware and software errors) are placed initially in in-core buffers and then transferred to a special file (a linked list of granules). This transfer is initiated whenever an error count threshold, or time limit is reached. The special file is then transferred to an ordered keyed file (ERRFILE) by the standard system ghost processor ERR:FIL which is automatically awakened by the system.

ERROR LOG LISTING

The keyed file (ERRFILE) may be listed and stored by the processor ELLA which allows a Customer Engineer to display and search the error file for patterns of errors to aid in preventive maintenance for 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.

REMOTE DIAGNOSTIC ASSISTANCE

On the Xerox 560, on-line diagnostics and certain on-line debugging processors (ANLZ, Delta, and ELLA) may be utilized via the Remote Assist Station (RAS) interface. After control is obtained from the local operator, Customer Engineers and/or diagnostic programmers at remote locations may access the system via this interface without interfering with the on-line COC users and without using any of the normal communication equipment. By evaluating the system under normal operating conditions, many software errors and hardware malfunctions may be detected and eliminated expeditiously with a minimum of computer down time.

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 1675. 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 only 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. Closes Transaction Processing common journal.
6. Saves in-core Transaction Processing files.
7. Packages error log.
8. Informs users of interruption.
9. Saves time, data, error log pointers, accounting information, symbiont file directory, public disk granule usage map, and executive communication.
10. 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 a recovery or single user abort, one of the recovery functions is to dump the contents of core memory into a special file in the :SYS account. This information is saved for later analysis by a system programmer using a special debugging program, ANLZ.

The ANLZ program may be called by the operator or system programmers to run as a privileged ghost, on-line, or batch job. The ANLZ program is also called automatically as a privileged ghost job by the recovery routine as one of the first jobs following a recovery or the first job following a single user abort. In any mode, ANLZ is completely command driven. It responds to commands that selectively display monitor tables, examine memory, and compare the dump with the running monitor. As a ghost job, ANLZ is an interactive program which communicates with the operator via the operator's console. (Reference: CP-V/SP Reference Manual, 90 31 13.)

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

MC – maintenance console (Xerox 560 Remote Assist Station)

A maximum of 13 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 2. 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 set to the value specified by the LIMIT control command.

Finally, these composite values are compared to the maximum values in the :USERS table or 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 (BDXX). If the limit is less than or equal to the system maximum (BMXX), 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 (BDXX), 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.

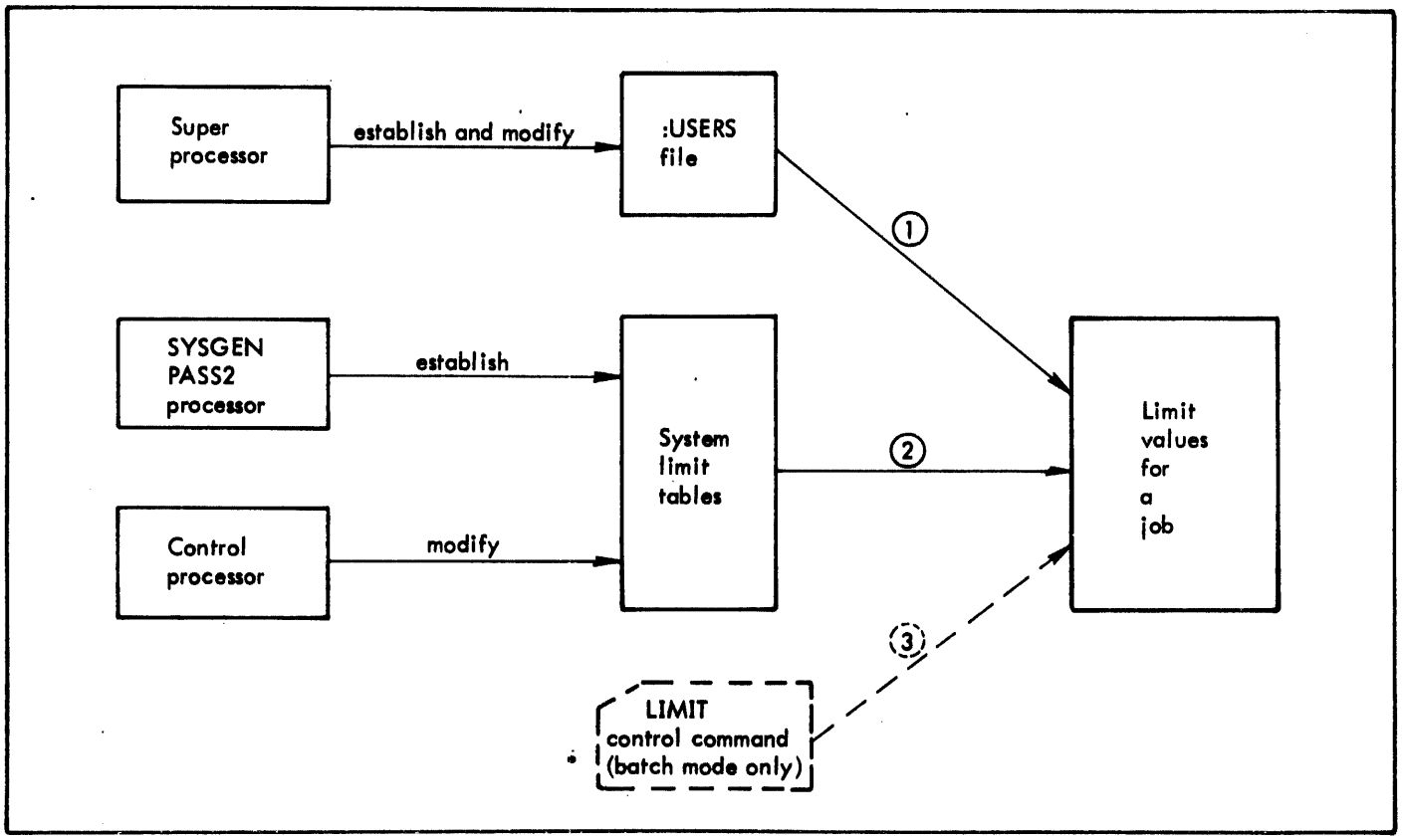


Figure 2. Establishing Limits for a Job

4. USER AUTHORIZATION

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 C0 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 is initially 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, either the initial log-on should specify a password or the TEL PASSWORD command should be used to establish a password for the system account after logging on.

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.

In addition, Super is used to create and maintain a file in the :SYS account called :PROCS. This file is keyed similarly to the :USERS file but does not necessarily contain a record for each user. The :PROCS file allows the system manager to restrict a user to an individually specified set of processors or to restrict an individual user from a specified set of processors. These restrictions may be individually controlled for the three modes of user access (on-line, batch, and ghost). The processors listed may be :SYS processors (both shared and unshared) and any executable load modules in any account.

Super is also used to create and maintain the :RBLOG file in the :SYS account 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 ten commands and associated options. The commands are

CREATE	P
MODIFY	FAST
DEFAULT	WORKSTATION
LIST	X
REMOVE	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

C[REATE] {account,name / account(name)}

where

account is the account under which the record is created. Account may be from 1 to 8 characters in length and may not contain any of the following characters:

| ; < > . / = ? and all control characters acted upon by COC.

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

{B / O / G}

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.)

Examples:

1. 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=5 (RET)
-- (RET)
-
```

2. Assume that a log-on file is to be created under account CLAS10 and name STUDENT. The user is only to have on-line access to EASY, BASIC, and FLAG and is to be restricted from executing any other processor.

```
-C CLAS10,STUDENT (RET)
--CALL=EASY (RET)
--XO=Y (RET)
--RP=EASY,0 (RET)
--RP=BASIC,0 (RET)
--RP=FLAG,0 (RET)
--PM=ALLOWED (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: | ; < > . / = ? and all control characters acted upon by COC. 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.
EX[PIRE] = {days,hours / NEVER / NONE}	"days,hours" specifies the default 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.

Table 5. CREATE Command Options (cont.)

Option	Description
{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. Default privilege for ghost jobs. B0 — Access and change the monitor. A0 — Read and write error file; request the devices; invoke diagnostics; authorize enqueue/dequeue automatically. 80 — Examine (but not change) the monitor. 40 — Default privilege level for batch and on-line.
{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.) If MC (maintenance console) is specified as a resource during a Super run, a special default :PROCS record is generated, replacing the current :PROCS record if there is one. The :PROCS record generated contains the following: PM=ALLOWED RP=ANLZ,O RP=DELTA,O RP=ELLA,O RP=OLTEST,O When MC is specified for a user, the P command and the XO, RP, and PM options are illegal. The MC option cannot be deleted. The only way to escape from the restrictions imposed by using the MC option is to use the Super REMOVE command and delete both the :USERS and :PROCS records for the particular user.
{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.

Table 5. CREATE Command Options (cont.)

Option	Description
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.
XO={Y[ES] / N[O]}	specifies whether or not this user is allowed to execute only processors in the :SYS account. The default is NO.
SE[CURITY]={Y[ES] / N[O]}	specifies whether or not a security check is to be performed on granules and core which will be used by this user. If YES, all core and granules that the user will access will be effectively erased before they may be accessed. The default is NO.
PM={A[LLOWED] / D[ISALLOWED]}	ALLOWED indicates that the :PROCS record for this user specifies processors that the user is allowed to use, all other processors being disallowed. DISALLOWED indicates that the :PROCS record for this user specifies processors that are disallowed, all other processors being allowed. This option is ignored by the system unless the RP option is also specified.
RP={processor-name / partial-name} [.account] [,R / ,[B][O][G]]	is the restricted processor option and adds, replaces, or removes the named processor in the :PROCS record. Super will respond with the message MODIFIED or REMOVED if it is not a new entry. The name must be from one to eleven characters in length. The account must be from one to eight characters in length. The mode of control (on-line, batch, and ghost) may be specified by the letters B, O, and G which are separated from the processor name by a comma. The letters are not separated. If mode flags are not specified (i.e., only a processor name is input), BOG is assumed. R specifies that the specified processor name is to be removed from the :PROCS record. A partial name is the first n characters of a processor name followed by a slash (/). When determining if a processor is allowed to execute, STEP will compare the processor name up to the slash. This allows a set of processors having identical leading characters to be restricted with one entry in the :PROCS record for a given user.

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 time (in minutes) for job execution time.
LO[t]	The number of pages of printed output from all shared processors involved in running a job.
PO[t]	The number of punched card records produced in running a job.
DO[t]	The number of pages of diagnostics produced in running a job.
UO[t]	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.

Table 6. System Services (cont.)

Service	Definition
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.

[t]A value of 32,767 (maximum) implies no limit.

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.

L[IST] {account,name[,name[,name]] / account(name[,name[,name]])}

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]

XO

SE[CURITY]

Mname

Pname

$

M

P

PL

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 PL is entered, a listing of the processor restriction list entries in the restricted processor file (:PROCS) record for the selected user(s) are output preceded by an indication of the mode of the list (allowed/disallowed). If there are no entries for the selected user(s), this is indicated. (If ALL has been specified, this null indication is omitted for improved output readability.)

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 (RET)
-- (RET)
```

(The names and account numbers are listed here.)

```
-
```

Example:

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

```
-CREATE 1234ABCD,C36 (RET)
--PASSWORD=SECRETX (RET)
--CALL=INITIAL.123ABC
--B$PRIV=40 (RET)
--O$PRIV=80 (RET)
--BM9T=2 (RET)
--BMSP=0 (RET)
--BPLP=Y (RET)
--OPLP=Y (RET)
--OMLO=50;BMLO=100 (RET)
--BMTIME=15 (RET)
--PM=ALLOWED (RET)
--RP=BASIC (RET)
--RP=EDIT (RET)
--XO=Y (RET)
-- (RET)
```

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

```
-LIST 1234ABCD,C36 (RET)
--ALL (RET)
-- (RET)
ID= 1234ABCD     C36
PA= SECRETX
CA= 123ABC       INITIAL
BI= 00  01  01
PR= 40  80  40
  M LO   100,50,0
  M TIME 15,0,0
  M 9T   2,0,0
  M SP   0,0,0
  P LP   Y,Y,N
XO=Y
SE=N
ALLOWED PROCESSORS:
 BOG BASIC
 BOG EDIT
```

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

R[EMOVE] {account,name / account(name)}

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. The REMOVE command removes both the log-on (:USERS) record and the associated restricted processors (:PROCS) record. An error message is given only if neither record exists.

Example:

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

```
:REMOVE 8634,ABLE (RET)
:
```

P The P command removes the specified user's record from the :PROCS file whether or not there exists a corresponding :USERS record. The format of the command is:

P{account,name / account(name)}

where

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

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

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, :PROCS and :RBLOG files will be kept open (and hence cannot 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.

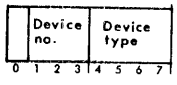
The LW option (which causes the options of a workstation or of all workstations to be listed) is a special case. When the LW option is specified, no other options may be specified. In fact, when the LW option is specified, Super outputs the requested listing without prompting for further WORKSTATION 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 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 3 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 cannot 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



where bit 0 is always set to one.

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.
RETRY = n	RBT, IRBT	Specifies, in decimal, the number of times that a failing operation is to be retried before the line is disconnected and the ERROR MAX message is output. Any number in the range 3 to 255 may be specified. The default value is 15.
{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 card punched on 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.
X3	IRBT	Specifies that this IRBT is capable of full multileaving. That is, it can accept records for multiple devices within the same block. X3 is the default between X3 and N3.
N3	IRBT	Specifies that this IRBT requires that each block contain records for only one device. A few IRBTs have this restriction.
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.
LOGON	IRBT	Specifies that the next line (or card) is the complete log-on record that will be transmitted to the master site when this slave CP-V system logs on. (When Super is called on-line, the record will be automatically blank padded to 80 characters.) LOGON is legal only for workstations for which SLV has been specified. It cannot be used with the RMT or RWSN options.
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.

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

Option	Type	Description
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.
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] = value	I,O	Specifies the subrecord control byte type for this device. The value may be P for printer type. C for card type. U for user supplied. X for special printer type. 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. SRCB=P and SRCB=X represent the two basic types of printer vertical format control. With SRCB=P, most space or skip operations are performed prior to the printing of the line as is common on Xerox printers. With SRCB=X, space or skip operations follow the print. Both types of vertical format control will work correctly for most IRBT printers, but print speed can often be improved by selecting the type which corresponds to the "natural mode" of the printer being used.
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.

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

Option	Type	Description
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 NB.
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 240. The default value is 80 unless LIST=Y or LIST=S is specified, in which case the default value is 132.
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 240. 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 240 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.

Table 9. Standard IRBT Device Option Values

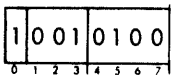
Device Type	RCB	IRCB	SRCB	SUSBIT
OC	91	92	C	40
CR	93	--	C	--
LP	94	--	P	800
CP	95	--	C	1

```
WORKSTATION id
     SYS
     NSYS
     RP
     CJOB
     RETRY
     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
          X3
          N3
          DD
          MST
          SLV
               RMT
               RWSN
               LOGON
          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 3. WORKSTATION Command Options

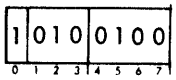
For example, if a system using standard RCBs has two line printers, the two RCBs would be

Line printer 1



(X'94')

Line printer 2



(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.

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(RET)
--TYPE=STND(RET)

-W STANDARD(RET)
--LW(RET)

ID=   4 STANDARD
TYP=IRBT
MODE=MST
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	94	P	0800	OUT	S	N	N	N	40	00	132	38
CP	95	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(RET)
--PASSWORD(RET)
--FLAGS(RET)
--$(RET)
--(RET)
.
.  (listing)
.
-REMOVE 8634,ABLE(RET)
-END(RET)
!
```

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 4 depicts a sample Super batch job deck.

Comment cards may appear anywhere in the Super input deck. They must have an asterisk in column one.

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

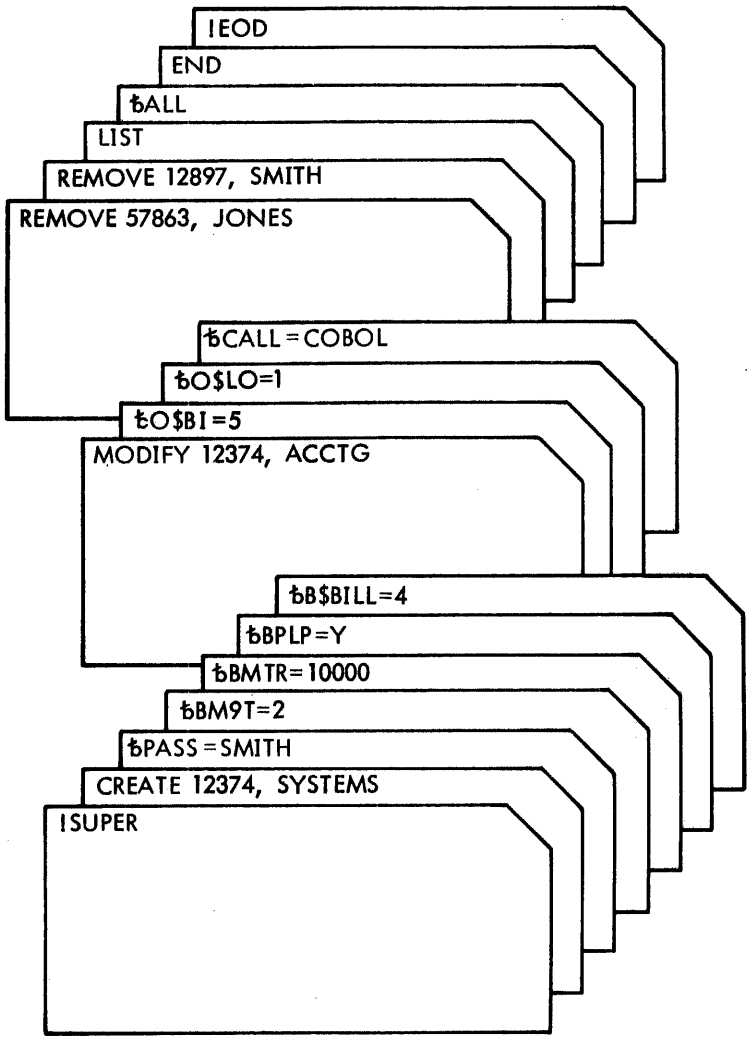


Figure 4. Sample Super Batch Job Deck

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.

GHOST OPERATION (SUPER)

It may be more convenient to make short Super runs operating in the ghost mode from the operator's console. This mode is initiated by the operator key-in

IGJOB SUPER

In this mode, commands must be typed in at the operator's console in the same format as for BATCH mode input. Additionally, the first line should contain only a new-line (NL) since it is ignored. This mode of operation is not recommended for lengthy sessions since it suspends operator messages. All output from running in ghost mode is directed to the line printer.

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.

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.
CAN'T GET PAGE FOR DEFAULT OPTION	There is no memory available to hold the internal record for the default command.
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 CHARACTER IN NAME/ACCT	An invalid character was detected while processing a restricted processor (RP) option.

Table 10. Super Error Messages (cont.)

Message	Description
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.
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.
OPTION DISALLOWED BECAUSE MC SPECIFIED	Specified command or option is not permitted when MC is set.
PROCESSOR NAME OR ACCOUNT TOO LONG	On a restricted processor (RP) option, either the name has exceeded 8 characters or the account has exceeded 12 characters.
:PROCS RECORD FULL. NAME REJECTED	There is no more space for restricted processor names in the :PROCS record. Perhaps clever use of the partial processor name feature will rectify this problem.

Table 10. Super Error Messages (cont.)

Message	Description
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 OO 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 ERROR IN COMMAND	There is a syntax error in the restricted processor (RP) option.
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.
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.

Table 10. Super Error Messages (cont.)

Message	Description
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} EX[PIRE]={days,hours / NEVER / NONE} {B / O / G} $BI[LLING] = charge (0 ≤ charge ≤ 7) {B / O / G} $PR[IVILEGE] = level where level can be one of the following hexadecimal codes: E0, C0, B0, A0, 80, 40. {B / O / G} Mname = {value / DELE[TE]}

Table 11. Super Command Summary (cont.)

Command	Description
C[REATE] {account,name / account(name)} (cont.)	{B / O / G} Pname = {Y[ES] / N[O] / DELE[TE]} XA[CCT] = character string XO {Y[ES] / N[O]} SE[CURITY] = {Y[ES] / N[O]} PM = {A[LLOWED] / D[ISALLOWED]} RP = {processor-name / partial-name/} [.account] [,R] [,[B] [O] [G]]
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] XO SE[CURITY] Mname Pname $ M P PL
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
P {account,name / account(name)}	Deletes a record from the :PROCS file. There are no options.
R[EMOVE] {account,name / account(name)}	Deletes a record from the log-on files. Also deletes the corresponding record from the restricted processor list file if it exists for this user. There are no options for this command.
W[ORKSTATION] id	Authorizes a workstation, specifies or changes 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 RETRY=n MRB NMRB NEM EM LPP=value MLP=value MCP=value MS[T] SL[V] DS[M]=mask X1 N1 X2 N2 X3

Table 11. Super Command Summary (cont.)

Command	Description
W[ORKSTATION]id (cont.)	N3 RM[T]=nn RW[SN]=xxxx LOGON DEV=devname DD=devname Device Options: IN OUT RC[B]=value IR[CB]=value SU[SBIT]=value SR[CB]=value 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, 15    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, 15    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 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 5 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. Items in the record with value of zero are suppressed from the printer.

In the accounting record, the start time and end time fields are expressed in minutes from midnight by default. However, these two fields may be expressed in seconds from midnight by patching bit 31 of S:OPTION to a one in the monitor. (S:OPTION is described in the CP-V Data Base Technical Manual, 90 19 95.)

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 6). 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 and requires at least C0 privilege. If the file :RATE does not exist, RATES creates it using the default value shown in Figure 6.
The RATES processor is called by entering the name of the processor in response to a TEL prompt for a command.

```
! RATES (RET)
=
```

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

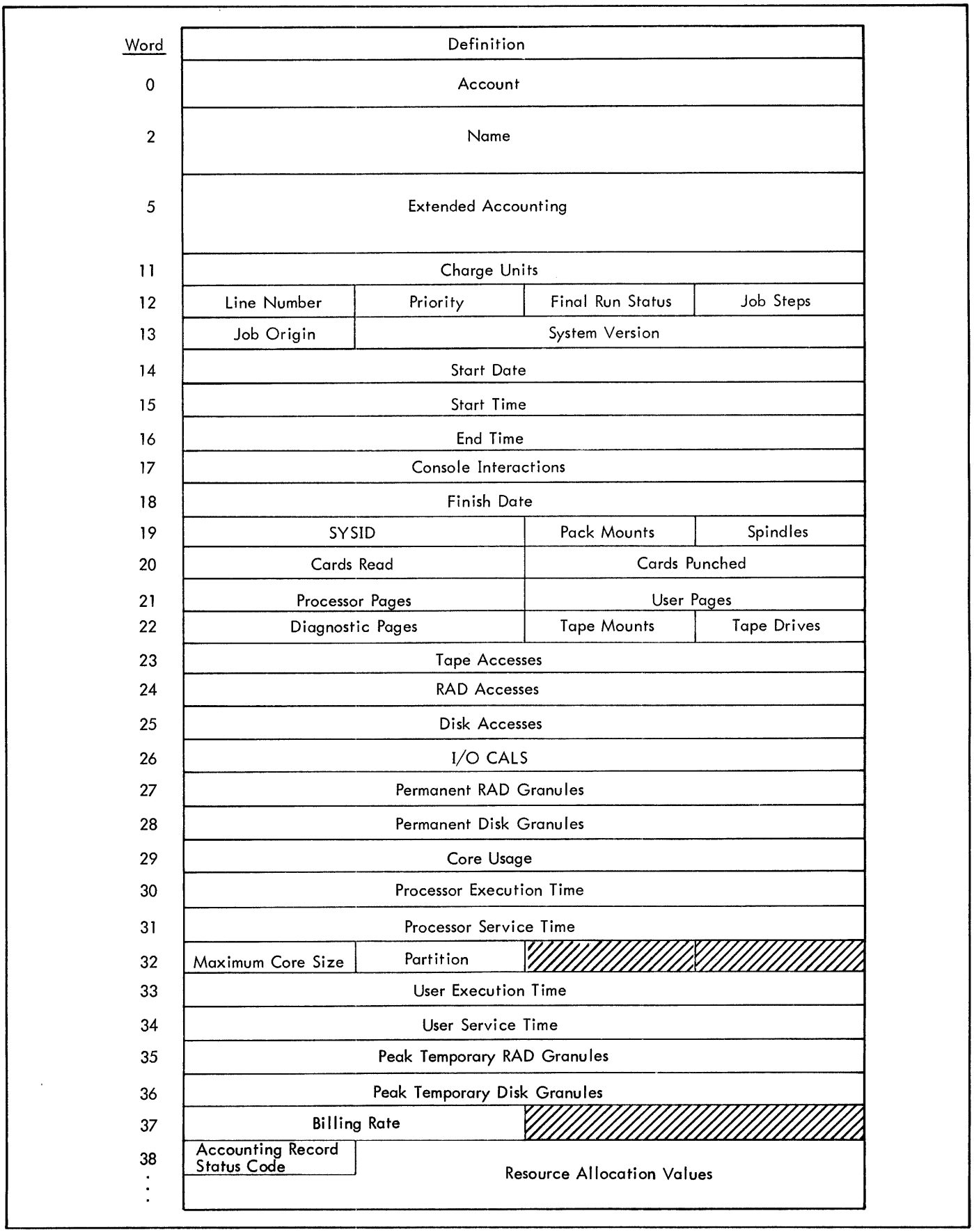


Figure 5. Structure of Accounting Record

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.
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 or terminal session in minutes (or, optionally, seconds) 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 of all types 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.
Tape Drives	Maximum number of tape drives of all types 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. These values include cooperative I/O.
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).
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 (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. These values represent maximum number of resources allocated to batch or available to on-line.

[t] One tick equals two milliseconds.

Table 13. Accounting Printout for Batch Jobs

Printed Format		Explanation
(Time and Date)		
ELAPSED JOB TIME	hh:mm:ss	Clock time in hours, minutes, and seconds for job or terminal session.
PARTITION NUMBER		Partition number under which the job ran.
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:	DRIVES ALLOCATED	xx	Number of tape drives allocated.
	TAPES MOUNTED	xx	Number of tapes mounted.
PACKS:	SPINDLES ALLOCATED	xx	Number of disk spindles allocated.
	PACKS MOUNTED	xx	Number of disk packs mounted.
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 permanent storage (in granules).
	NUMBER OF SWAPS	xxxx	Number of times the user was swapped.
	RESOURCES ALLOCATED CO=xx 9T=xxxx 7T=xxxx	(etc.)	Values of resources allocated.
	CHARGE UNITS	xxxxxxxx	Total charge units.

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]

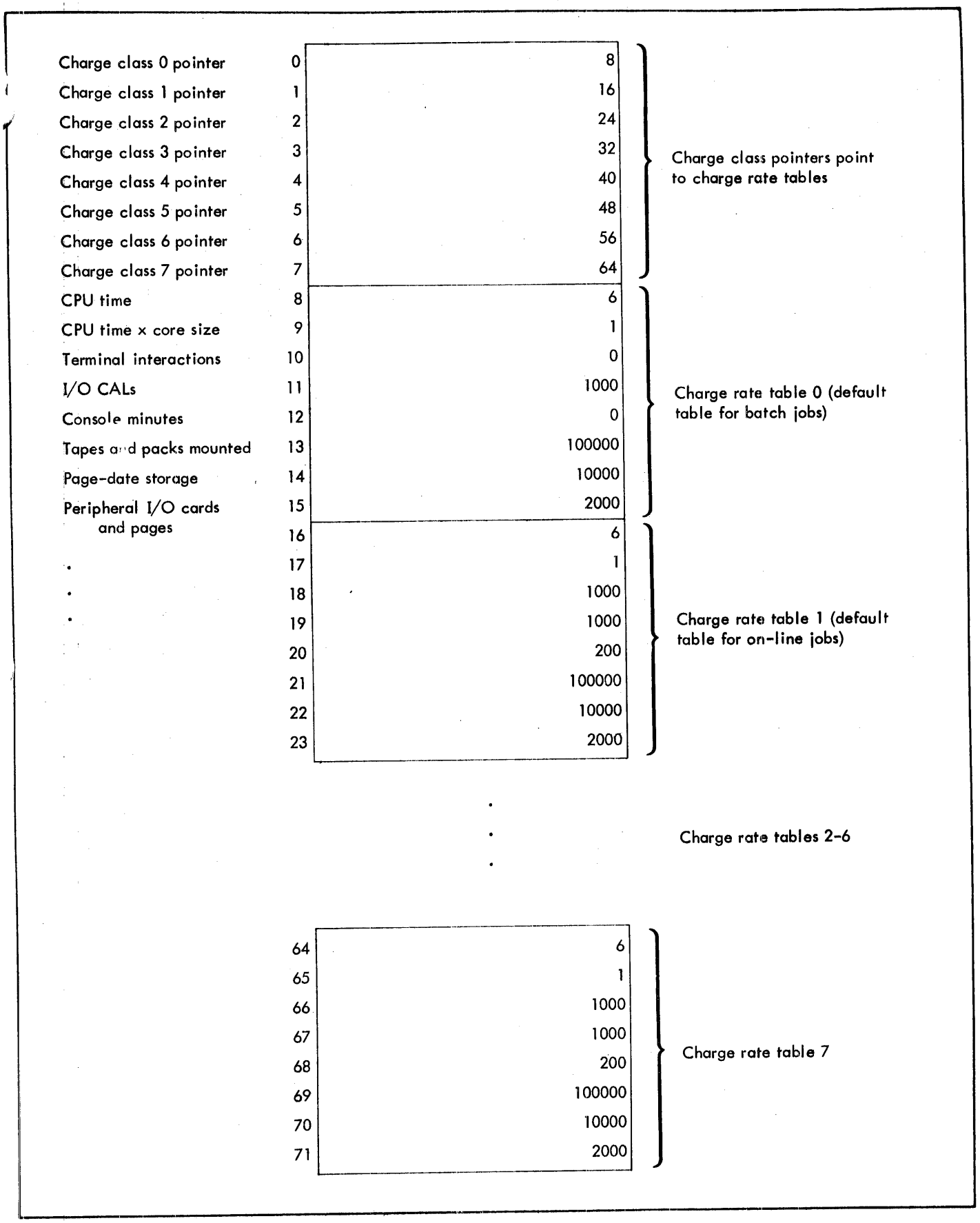


Figure 6. Structure of :RATE File

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 $0 \leq table \leq 7$.

value specifies the units to be charged for a particular item and has a range $0 \leq value \leq 99999999$. 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 $0 \leq table \leq 7$.

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.

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

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 are described and listed. Several of these display items are utilized in both Control and STATS operations. Other groups of display items (applicable to one processor only) are discussed in the section about the particular processor.

The display items described 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 a means of 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 parameters that affect system performance.

A user privilege of 80 is required for displaying information. A privilege of C0 is required for modifying the system parameter limits found in Table 16 and the partition attributes found in Table 18.

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 columns of the card. A blank card should be used after the last item in an ADD or DROP sequence.

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. Exceptions to this are the CONTROL! and the partition display commands which are output to the line printer.

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			
BUN	S:BUAIS	Maximum number of concurrent batch users.	users	0	# of batch① partitions or SMUIS-OUM-S:GUAIS (whichever is smaller)
OUM	S:OUAIS	Maximum number of on-line users allowed in the system.	users	0	SMUIS-BUM-S:GUAIS
		Execution Control			
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	30,000
QMIN	SL:QMIN	Amount of uninterrupted compute time guaranteed a user after selection.②	msecs	0	30,000
SQUAN	SL:QUAN	Amount of time a user is guaranteed core residency before swap out.②	msecs	0	30,000
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
RETIM	SL:RET	Amount of time an on-line user's swap image is to be retained after the user has been disconnected due to a line hang up condition.	minutes	0	65,535
BSTRT	SL:BSTRT	Slave processor auto-start inhibit. BSTRT = 1 inhibits; BSTRTR = 0 permits.	numeric	0	1
MAXQn	SH:MAXQ	Maximum compute interval for a single job for slave processor. The n value (1, 2, or 3) specifies the slave CPU number.	msecs	MINQn	32,767
MINQn	SH:MINQ	Interrupt value for slave processor. The n value (1, 2, or 3) specifies the slave CPU number.	msecs	10	MAXQn
MPCALR	SL:MPCALR	Compute per CAL threshold. If the user's average time between CALs is less than this value, the user will not be scheduled to a secondary processor.	msecs	0	32,767
OIOTA	SL:OIOTA	The amount of TICs that are deducted from the user's current quantum for each I/O.	tics	0	32,767
BIOTA	SL:BIOTA	The amount of TICs that are deducted from the user's current quantum for each I/O.	tics	0	32,767
GIOTA	SL:GIOTA	The amount of TICs that are deducted from the user's current quantum for each I/O.	tics	0	32,767

Table 16. Control Parameters (cont.)

Control Name	System DEF	Description	Unit	Minimum Value	Maximum Value
		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 seconds of output at which to block terminal output. ④	seconds	UB	32,767
UB	SL:UB	Number of seconds of output at which to unblock terminal output. ⑤	seconds	1	TB
ONCB	SL:ONCB	Maximum number of COC buffers allowed per user.	buffers	2	255
OLTO	SL:OLTO	Log-on time out. (No response to the log-on salutation causes the line to be hung up by the system.)	minutes	1	32,767
OITO	SL:OITO	Terminal input time out. (No response to terminal read causes logg off and line hang up.)	minutes	1	32,767
RAM	SL:RAMR	Maximum number of concurrent read-ahead operations in the system.	numeric	0	RASIZE
RATO	SL:RATOR	Time after which a read-ahead operation will be aborted.	msecs	0	32,767
CHUNK	SL:COMAXG	The size of a concurrent output mode file chunk. Jobs running in concurrent output mode require one output symbiont file slot for every CHUNK granules of output.	granules	5	250
AIRTO	SL:AIRTO	Time after which unused AIR block will be purged.	msecs	0	500,000
AIRM	SL:AIRM	Maximum number of AIR granules to be held in core at one time.	numeric	0	RASIZE
		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

Table 16. Control Parameters (cont.)

Control Name	System DEF	Description	Unit	Minimum Value	Maximum Value
		Exit Control Processing Limits (cont.)			
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			
Tres (6) (12)	SH:RTOT	Total resource available for all jobs.	(7)	0	System Capability
BTres (6) (12)	SH:RBSUM	Total resource available for all batch jobs.	(7)	0	SH:RTOT
OTres (6) (12)	SH:ROSUM	Total resource available for all on-line jobs. (Meaningless for spindles.)	(7)	0	SH:RTOT
GTres (6) (12)	SH:RGSUM	Total resource available for all ghost jobs.	(7)	0	SH:RTOT
BMres (6) (12)	SB:RBMX	Maximum value of a resource that can be requested by a batch job.	(7)	0	SH:RBSUM
OMres (6)	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
GMres (6)	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
BDres (6)	SB:RBDF	Default value of a resource that is allocated to a batch job.	(7)	0	SB:RBMX
ODres (6)	SB:RODF	Default value of a resource that is allocated to an on-line job, if not specified in :USERS record.	(7)	0	SB:ROMX
GDres (6)	SB:RGDF	Default value of a resource that is allocated to a ghost job.	(7)	0	SB:RGMX
		Service Limit Control			
BMserv (8)	SL:BMX	Maximum value of a service that can be requested by a batch job.	(9)	0	32,767
OMserv (8)	SL:OMX	Maximum value of a service that can be requested by an on-line job.	(9)	0	32,767
GMserv (8)	SL:GMX	Maximum value of a service that can be requested by a ghost job.	(9)	0	32,767
BDserv (8)	SL:BDF	Default value of a service that can be assigned to a batch job.	(9)	0	SL:BMX
ODserv (8)	SL:ODF	Default value of a service that can be assigned to an on-line job.	(9)	0	SL:OMX
GDserv (8)	SL:GDF	Default value of a service that can be assigned to a ghost job.	(9)	0	SL:GMX

Table 16. Control Parameters (cont.)

Control Name	System DEF	Description	Unit	Minimum Value	Maximum Value
		Symbiont and Special Feature Limit Control			
BDfa (10)	S:SYMDB	Default authorization of a symbiont device or a special feature for a batch job.	-	0 (11)	1 (11)
ODfa (10)	S:SYMDO	Default authorization of a symbiont device or a special feature for an on-line job.	-	0 (11)	1 (11)
GDfa (10)	S:SYMDG	Default authorization of a sumbiont device or a special feature for a ghost job.	-	0 (11)	1 (11)

(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. This 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. X'FE' is the recommended value.

(4) TB is the number of seconds of output at which the user is swapped out to the swapping disk.

(5) UB is the number of seconds of output remaining 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 resource names are

CO - core
9T - 9-track tapes
7T - 7-track tapes
SP - disk pack spindles
MC - Maintenance Console (Xerox 560 Remote Assist Station)

(7) Unit depends on the particular resource. Units for standard resources are "tape drivers" 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.

(12) The intent of these values for sharable resources (disk packs) may not be clear and is therefore explained here. Once a pack is in use, other users of any type (batch, on-line, or ghost) can use the pack without additional allocation of system resources; that is, OCxx, BCxx, and GCxx do not change. The user must be authorized to use the pack via ODxx, BDxx, or GDxx in CONTROL or via OMxx, ODxx, or GMxx in SUPER. To prevent on-line users from using private packs (during part of the day), all users can be authorized via OMxx=0 in SUPER and ODxx=0 for no access or ODxx=non-zero for access in CONTROL.

Table 17. Current System Values

Control Name	System DEF	Description	Units
BCres	SH:RBCU	Current value of a resource allocated to all batch jobs.	-
OCres	SH:ROCU	Current value of a resource allocated to all on-line jobs.	-
GCres	SH:RGCU	Current value of a resource allocated to all ghost jobs.	-
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

(This page intentionally left blank.)

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
TIME③	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
ACCT④	PLD:ACT	X	Account number of job currently executing in partition.	EBCDIC	-	-
CUR④	PLH:CUR	X	Number of jobs that have been run under current definition of this partition.⑤	jobs	0	32,767
TOL④	PLH:TOL	X	Total number of jobs run under this partition since system startup.⑤	jobs	0	32,767
HOLD	PLH:FLG⑥ (bit 0)		Core residency control flag. 1 → hold job in core. 0 → allow swap out.	boolean	0 or 'NO'	1 or 'YES'
LCL	PLH:FLG (bit 12)		Job Control for following types of jobs: LCL = local card reader. TRM = jobs entered via JOBENT (IBATCH). RB = remote batch processing.	boolean	0 disallow	1 allow
TRM	PLH:FLG (bit 13)			boolean	0 disallow	1 allow
RB	PLH:FLG (bit 14)			boolean	0 disallow	1 allow
LOCK	PLH:FLG (bit 15)		Partition selection lock flag. 0 → select jobs for this partition. 1 → lock partition from further selection.	boolean	0 or 'NO'	1 or 'YES'
USER④	PLB:USR	X	ID number of user currently executing in partition.	hexadecimal	0	X'FF'
SP①	PLB:MAX⑥		Maximum number of spindles that may be used for selection under this partition.	spindles	0	SH:ROT
7T①	PLB:MAX		Maximum number of 7-track drives that may be used for selection under this partition.	7T drives	0	SH:ROT
9T①	PLB:MAX		Maximum number of 9-track drives that may be used for selection under this partition.	9T drives	0	SH:ROT
CO①	PLB:MAX		Maximum size of core that may be used for selection under this partition.	K words	0	SH:ROT
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 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 resource names are

CO - core
9T - 9-track tapes
7T - 7-track tapes
SP - disk pack spindles
MC - Maintenance Console (Xerox 560 Remote Assist Station)

② Unit depends on the particular resource. Units for standard resources are "tape drives" for tapes, "spindles" for disk packs, and "K words" for core.

③ 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.

④ ACCT, CUR, TOL, and USER may be displayed but not modified.

⑤ Not including the currently executing job.

⑥ 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
5. MOS Memory Correctable Error Commands
   - MOS
   - THRESHOLD
   - CURRENT
   - RCE
   - END

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. A similar sub-command level is provided for MOS memory control and display.

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 ⏎

-AD ⏎

-ADXYZ⏎

-ADD ITEMS TO PRINT LIST ⏎

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 ⏎ 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 ⏎

.QMIN ⏎

.OM7T ⏎

. ⏎

-

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 ⏎ alone after the prompt for input.

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

-DROP ⏎

.QMIN ⏎

.OM7T ⏎

. ⏎

-

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 ⏎

.ALL ⏎

. ⏎

-LIST ⏎

-AD ⏎

.QUAN ⏎

.TB ⏎

.UB ⏎

.BCSP ⏎

. ⏎

-DROP ⏎

.TB ⏎

.UB ⏎

. ⏎

-LIST ⏎

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 ⏎

.ALL ⏎

. ⏎

-ADD ⏎

.OMCO ⏎

.OUM ⏎

. ⏎

-CONTROL ⏎

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 7.

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 ⏎

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 ⏎

-QUAN ⏎

QUAN = 450

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

```
MAX #USERS IN SYS          (UM)=   106    CURRENT # USER IN SYS    (UC)=    19
MAX # BATCH USERS          (BUM)=   16    CURRENT # BATCH USERS    (BUC)=     1
MAX # ONLINE USERS         (OUM)=   75    CURRENT # ONLINE USERS   (OUC)=    13
MAX # GHOST JOBS           (GUM)=   15    CURRENT # GHOST JOBS     (GUC)=     5
PHYSICAL WORK PAGE MAX     (PWP)=   20    MSEC ONLINE QUANTUM      (QUAN)=  500
PRIORITY INCREMENT          (PI)=    0    ONLINE MINIMUM QUAN      (QMIN)=    4
MAX BATCH I/O COUNT        (BXMF)=    6    MAX ONLINE I/O COUNT     (OXMF)=    6
MIN BATCH I/O COUNT        (BIMF)=    3    MIN ONLINE I/O COUNT     (OIMF)=    3
ELIM DEFAULT - PSTORE      (EPS)=   100    MSEC CORE-RESIDENCY      (SQUAN)=  400
ELIM DEFAULT - TIME        (ETIME)=  30    LOG-ON TIME-OUT          (OLTO)=    3
ELIM DEFAULT - LO          (ELO)=   10    TERM INPUT TIME-OUT      (OITO)=   30
ELIM DEFAULT - PO          (EPO)=   50    # CHARS TERMINAL BLOCK    (TB)=   240
ELIM DEFAULT - DO          (EDO)=   10    # CHARS TERM UNBLOCK      (UB)=    30
ELIM DEFAULT - UO          (EUO)=   10    MAX # COC BUFS/USER      (ONCB)=   25
ELIM DEFAULT - TSTORE      (ETS)=   64    BASE EXEC PRIO-BATCH     (BPRIO)=  254
BASE EXEC PRIO-ONLINE      (OPRIO)=  254  BASE EXEC PRIO-GHOST     (GPRIO)=  254
MAXIMUM # READ-AHEADS      (RAM)=    6    READ-AHEAD TIME-OUTS     (RATO)= 5000
MAX # AIR ENTRIES          (AIRM)=   8    AIR TIME-OUT             (AIRTO)=*****
RETENTION TIME             (RETIM)=  0    SYMBIONT OVERFLOW         (PFA)=    1
COMODE BLOCKING SIZE       (CHUNK)=  40
SLV CPU ALTO INHIBIT       (BSTRT)=  0    MP CAL RATE THRESHLD     (MPCALR)=    0
SLV CPU MAX QUAN           (MAXQ1)= 5000  SLV CPU MIN QUAN         (MINQ1)=  200
SLV CPU MAX QUAN           (MAXQ2)= 5000  SLV CPU MIN QUAN         (MINQ2)=  200
SLV CPU MAX QUAN           (MAXQ3)= 5000  SLV CPU MIN QUAN         (MINQ3)=  200

 *** ==> NUMERIC DISPLAY OVERFLOW
AIR TIME-OUT               (AIRTO)=160000
SYSTEM TOTAL RESOURCE CO     =32767    SYSTEM TOTAL RESOURCE QT     =    6
SYSTEM TOTAL RESOURCE SP     =    3    SYSTEM TOTAL RESOURCE SQ     =   16
TOTAL *GHOST*  RSRCE  TCO    =32767    TOTAL *GHOST*  RSRCE  TQT    =    8
TOTAL *GHOST*  RSRCE  TSP    =    4    TOTAL *GHOST*  RSRCE  TSQ    =   16
CURNT *GHOST*  RSRCE  CCO    =    0    CURNT *GHOST*  RSRCE  CQT    =    0
CURNT *GHOST*  RSRCE  CSP    =    0    CURNT *GHOST*  RSRCE  CSQ    =    0
MAX GHOST  JOB RSRCE  MCO    =  127    MAX GHOST  JOB RSRCE  MQT    =    8
MAX GHOST  JOB RSRCE  MSP    =    4    MAX GHOST  JOB RSRCE  MSQ    =   15
DFT GHOST  JOB RSRCE  DCO    =   64    DFT GHOST  JOB RSRCE  DQT    =    1
DFT GHOST  JOB RSRCE  DSP    =    1    DFT GHOST  JOB RSRCE  DSQ    =   15
TOTAL *ONLINE* RSRCE  TCO    =32767    TOTAL *ONLINE* RSRCE  TQT    =    6
TOTAL *ONLINE* RSRCE  TSP    =    4    TOTAL *ONLINE* RSRCE  TSQ    =   16
CURNT *ONLINE* RSRCE  CCO    =    0    CURNT *ONLINE* RSRCE  CQT    =    1
CURNT *ONLINE* RSRCE  CSP    =    0    CURNT *ONLINE* RSRCE  CSQ    =    0
MAX ONLINE JOB RSRCE  MCO    =  127    MAX ONLINE JOB RSRCE  MQT    =    8
MAX ONLINE JOB RSRCE  MSP    =    4    MAX ONLINE JOB RSRCE  MSQ    =   15
DFT ONLINE JOB RSRCE  DCO    =   45    DFT ONLINE JOB RSRCE  DQT    =    0
DFT ONLINE JOB RSRCE  DSP    =    0    DFT ONLINE JOB RSRCE  DSQ    =    0
TOTAL *BATCH*  RSRCE  TCO    =  350    TOTAL *BATCH*  RSRCE  TQT    =    6
TOTAL *BATCH*  RSRCE  TSP    =    4    TOTAL *BATCH*  RSRCE  TSQ    =   16
CURNT *BATCH*  RSRCE  CCO    =   16    CURNT *BATCH*  RSRCE  CQT    =    0
CURNT *BATCH*  RSRCE  CSP    =    0    CURNT *BATCH*  RSRCE  CSQ    =    0
MAX BATCH  JOB RSRCE  MCO    =   64    MAX BATCH  JOB RSRCE  MQT    =    6
MAX BATCH  JOB RSRCE  MSP    =    2    MAX BATCH  JOB RSRCE  MSQ    =    0
DFT BATCH  JOB RSRCE  DCO    =   24    DFT BATCH  JOB RSRCE  DQT    =    0
DFT BATCH  JOB RSRCE  DSP    =    0    DFT BATCH  JOB RSRCE  DSQ    =    0
MAX GHOST  JOB SRVCE  MTIME  = 9999    MAX GHOST  JOB SRVCE  MLO    = 9999
MAX GHOST  JOB SRVCE  MPO    = 9999    MAX GHOST  JOB SRVCE  MDO    = 9999
MAX GHOST  JOB SRVCE  MLO    = 9999    MAX GHOST  JOB SRVCE  MPSTO  = 1000
MAX GHOST  JOB SRVCE  MTSTO  = 1000    MAX GHOST  JOB SRVCE  MFPOO  =    6
MAX GHOST  JOB SRVCE  MTDIS  =32767    MAX GHOST  JOB SRVCE  MPDIS  =32767
DFT GHOST  JOB SRVCE  DTIME  = 9999    DFT GHOST  JOB SRVCE  DLO    = 9999
DFT GHOST  JOB SRVCE  DPO    = 9999    DFT GHOST  JOB SRVCE  DDO    = 9999
DFT GHOST  JOB SRVCE  DLO    = 9999    DFT GHOST  JOB SRVCE  DPSTO  =   64
DFT GHOST  JOB SRVCE  DTSTO  =   64    DFT GHOST  JOB SRVCE  DFPOO  =    6
DFT GHOST  JOB SRVCE  DTDIS  =32767    DFT GHOST  JOB SRVCE  DPDIS  =32767
MAX ONLINE JOB SRVCE  MTIME  =  999    MAX ONLINE JOB SRVCE  MLO    = 9999
MAX ONLINE JOB SRVCE  MPO    = 9999    MAX ONLINE JOB SRVCE  MDO    = 9999
MAX ONLINE JOB SRVCE  MLO    = 9999    MAX ONLINE JOB SRVCE  MPSTO  =  100
MAX ONLINE JOB SRVCE  MTSTO  = 1000    MAX ONLINE JOB SRVCE  MFPOO  =   18
MAX ONLINE JOB SRVCE  MTDIS  = 1000    MAX ONLINE JOB SRVCE  MPDIS  = 1000
DFT ONLINE JOB SRVCE  DTIME  =  999    DFT ONLINE JOB SRVCE  DLO    =  499
DFT ONLINE JOB SRVCE  DPO    = 9999    DFT ONLINE JOB SRVCE  DDO    = 9999
```

Figure 7. Example of a CONTROL! Display

```
DFT ONLINE JOB SRVCE  DUØ    = 9999   DFT ONLINE JOB SRVCE  DPSTØ  =   50
DFT ONLINE JOB SRVCE  DTSTØ  = 1000   DFT ONLINE JOB SRVCE  DFPØØ  =    4
DFT ONLINE JOB SRVCE  DTDIS  = 1000   DFT ONLINE JOB SRVCE  DPDIS  =  100
MAX BATCH  JOB SRVCE  MTIME  =  999   MAX BATCH  JOB SRVCE  MLØ    = 9999
MAX BATCH  JOB SRVCE  MPØ    = 9999   MAX BATCH  JOB SRVCE  MDØ    =  999
MAX BATCH  JOB SRVCE  MUØ    = 9999   MAX BATCH  JOB SRVCE  MPSTØ  =   50
MAX BATCH  JOB SRVCE  MTSTØ  = 1000   MAX BATCH  JOB SRVCE  MFPØØ  =   18
MAX BATCH  JOB SRVCE  MTDIS  = 1000   MAX BATCH  JOB SRVCE  MPDIS  = 1000
DFT BATCH  JOB SRVCE  DTIME  =   10   DFT BATCH  JOB SRVCE  DLØ    =  499
DFT BATCH  JOB SRVCE  DPØ    =   99   DFT BATCH  JOB SRVCE  DDØ    =   50
DFT BATCH  JOB SRVCE  DLØ    =  250   DFT BATCH  JOB SRVCE  DPSTØ  =    0
DFT BATCH  JOB SRVCE  DTSTØ  = 1000   DFT BATCH  JOB SRVCE  DFPØØ  =    8
DFT BATCH  JOB SRVCE  DTDIS  = 1000   DFT BATCH  JOB SRVCE  DPDIS  =  100
SYM/FAUTH DFT GHOST  DCR    =    1   SYM/FAUTH DFT GHOST  DLP    =    1
SYM/FAUTH DFT GHOST  DCP    =    1   SYM/FAUTH DFT GHOST  DAS    =    1
SYM/FAUTH DFT GHOST  DWC    =    1   SYM/FAUTH DFT GHOST  DEQ    =    1
SYM/FAUTH DFT GHOST  DJE    =    1   SYM/FAUTH DFT GHOST  DRP    =    1
SYM/FAUTH DFT GHOST  DTP    =    1   SYM/FAUTH DFT ONLINE DCR    =    1
SYM/FAUTH DFT ONLINE DLP    =    1   SYM/FAUTH DFT ONLINE DCP    =    1
SYM/FAUTH DFT ONLINE DAS    =    1   SYM/FAUTH DFT ONLINE DWC    =    1
SYM/FAUTH DFT ONLINE DEQ    =    1   SYM/FAUTH DFT ONLINE DJE    =    1
SYM/FAUTH DFT ONLINE DRP    =    1   SYM/FAUTH DFT ONLINE DTP    =    1
SYM/FAUTH DFT BATCH  DCR    =    1   SYM/FAUTH DFT BATCH  DLP    =    1
SYM/FAUTH DFT BATCH  DCP    =    1   SYM/FAUTH DFT BATCH  DAS    =    1
SYM/FAUTH DFT BATCH  DWC    =    1   SYM/FAUTH DFT BATCH  DEQ    =    1
SYM/FAUTH DFT BATCH  DJE    =    1   SYM/FAUTH DFT BATCH  DRP    =    1
SYM/FAUTH DFT BATCH  DTP    =    1
END
```

Figure 7. 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 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 processes 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 and an appropriate partition are available. If the job fits into more than one available partition, it is put into the highest numbered such partition.

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 the various partitions 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 exclusive 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 effects 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 10, Example of Partition Commands.

DISPLAY n This command displays all of the attributes of partition n that have their print flags on. The format of the command is

$$\text{DI[SPLAY]} \left\{ \begin{matrix} n \\ n_1 - n_2 \\ \text{ALL} \end{matrix} \right\}$$

where

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

$n_1 - n_2$ is a range of batch partition numbers.

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

In the example Figure 8, 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.

Figure 9 is the default display, i.e., the partition display before any items are ADDed or DROPed. All resource attributes available in the running system are automatically included.

```
≥DISPLAY 1

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 8. Example of the DISPLAY Command

```
≥DISPLAY 2
 PART LOCK HOLD LOL  TRM  RB   TIME   CO    9T   7T   SP   SQ
   2   NO   NO  YES  YES  YES  0-300  1-24  0-4  0-1  0-0  1-1
≥
```

Figure 9. Default Partition Display Example

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

$$\left\{\begin{matrix} n \\ n_1 - n_2 \\ ALL \end{matrix}\right\} attribute$$

where

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

$n_1 - n_2$ 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 (RET)
 12 QUAN = 400
>
```

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

Example:

```
>ALL 7T (RET)
 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 further on. Note that the only blank allowed is between the n and attribute specifications.

$$\left\{\begin{matrix} n \\ n_1 - n_2 \\ ALL \end{matrix}\right\} attribute = number$$

where

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

$n_1 - n_2$ 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 (RET)
 5 SP=0-2
>5 SP=2-2 (RET)
>5 SP (RET)
 5 SP=0-2
>STORE (RET)
 VALUES STORED
>5 SP (RET)
 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 2-2 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, a flag is set for MBS. On the next MBS processing cycle, 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 then cleared; therefore, no new attribute values are entered in the tables when the STORE command is used. 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, CONTROL1, 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 10 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 partition 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 11.

```
>10 TIME=0-4↵
>3 USER#↵
 3 USER#=24
>↵
←
-LIST↵
 BUC
```

Figure 10. 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 10. 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 11. 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 11. Example of Partition Commands (cont.)

MOS MEMORY COMMANDS

These commands control and display MOS memory single-bit correctable error (SBCE) logging parameters. The commands enable or disable the hardware reporting and software logging of any SBCEs, and are used to establish a threshold count of SBCEs for the memory banks. If the threshold is exceeded, reporting is disabled. The MOS commands can be used to display the current controls and the counts of SBCEs per bank of memory. When no MOS commands are given to Control, defaults are established during system initialization for each MOS memory bank as follows:

Threshold = 50
REC = ON

System Recovery saves and then restores the current MOS controls. Therefore, it is advisable to run the Control job that contains the MOS commands on a daily basis to establish a sampling of any SBCEs.

MOS This command passes execution to the MOS sub-command level. If the user has at least privilege 80, this command can display the current MOS memory error reporting data. If the user has at least privilege C0, this command can display and change the MOS error reporting data. This command must be given before any MOS-associated commands will be accepted. The format for the command is

MO[S]

Control then prompts for MOS-associated commands. An example of the MOS command is given in Figure 11.1.

CURRENT This command displays the current number of single-bit correctable errors (SBCE) that have been reported on a memory bank, the current threshold value, and whether the Report Correctable Errors (RCE) flag is on or off. The format of the command is

CU[RRENT] [ALL | bank | unit | unit,bank]

where

ALL — specifies that the status of all MOS memory banks in operation are to be reported.

bank — specifies the bank to be reported. Possible values are the letters A through D. If bank is specified without unit, unit 0 is assumed.

unit — specifies the memory cabinet to be reported; all active banks in the unit are reported. Possible values are 0 and 1.

unit,bank — specifies that the selected bank in the selected unit is to be reported.

```
!CONTROL

-MOS

<CURRENT A
 UNIT Ø BANK A  COUNT= 5Ø  THRESHOLD= 5Ø  RCE=OFF

<THRESHOLD A=1ØØ
 UNIT Ø BANK A  COUNT=  Ø  THRESHOLD=1ØØ  RCE=ON

<RCE OFF,B
 UNIT Ø BANK B  COUNT= 16  THRESHOLD= 5Ø  RCE=OFF
<END

-END
```

Figure 11.1 Sample of MOS Command and MOS Sub-commands.

THRESHOLD This command allows the system manager to set the number of SBCEs that are reported to ERR:FIL. When the threshold value on a particular bank is reached, the reporting mechanism in that bank is disabled. The form of the command is

TH[RESHOLD] [ALL / unit / bank / unit,bank] = n[,NO]

where

ALL	specifies that the threshold value on ALL banks is to be set to n.
unit	specifies that the threshold value is to be set to n for the specified unit. Possible values are 0 and 1.
bank	specifies that the threshold is to be set to n for the specified bank. Possible values are the letters A-D. If unit is not specified, unit 0 is assumed.
unit,bank	specifies that the threshold value is to be set to n on the specified bank (A-D) in the specified unit (0-1).
n	is the number of errors to report before RCE is set off for the selected elements. $0 \leq n \leq 255$.
NO	specifies that the "current" value is not to be chaned for the specified bank/unit. If not specified, the current count is set to zero.

In any case, an attempt is made to turn RCE ON after the threshold value is changed. However, if the threshold value is less than or equal to the current value, RCE is not activated on the bank that fails the comparison.

RCE This command specifies whether or not SBCEs are to be reported. The form of the command is

RCE [ON / OFF] [,ALL / ,bank / ,unit / ,unit,bank]

where

ON	specifies that SBCEs are to be reported on the specified element.
OFF	specifies that SBCEs are not to be reported on the specified element.
ALL, bank, unit, unit,bank	Are the same as described for the THRESHOLD command.

Execution of this command prints a CURRENT status for the specified elements. Note that setting RCE on for an element when its current value is equal to or greater than its threshold value results in RCE OFF for that element.

END This command specifies that control is to exit the MOS sub-command level and return to the command level of control.

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 10.

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 11.

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

TI[ME]

Example:

-TIME ↵

12:32

SAMPLE COMMAND SEQUENCE

The command sequence in Figure 12 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 that item display flags be set)
.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 12. 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 while reading through the M:SI DCB or writing through 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 C0	The user does not have privilege level of at least X'C0' 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 display monitor tables.
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 or to ADD or DROP a display item that does not exist or is incorrectly represented.
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.
SYSTEM NOT GEN'ED FOR THAT CONTROL NAME	An attempt was made to change or to display a control parameter for which the running system was not SYSGENed.
****	A numeric overflow occurred on CONTROL or CONTROLI display of a system control parameter. The parameter will automatically be displayed as a single parameter display.

Table 22. Control Command Summary (Control Level)

Command	Description
-AD[D] .item .item ⋮	Turns on the print flags for the specified items. Any item may be the Control Name of any item in Tables 16 and 17.
BREAK (i.e., BRK)	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 on 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.
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.
MO[S]	Turns control over to the MOS memory sub-command display (with privilege of at least X'80') or modification (with at least X'C0').
TH[RESHOLD] [ALL / unit / bank / unit,bank] = n[,NO]	Sets the threshold value of the selected MOS memory element to n. If NO is specified, the current count is not changed.
CU[RRENT] [ALL / bank / unit / unit,bank]	Displays the current MOS memory parameters for the selected elements.
RCE [ON / OFF] [,ALL / ,bank / ,unit / ,unit,bank]	Sets the SBCE reporting for the specified element(s) ON or OFF.

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 Table 18.
BREAK (i.e. ⓫)	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.
DI[SPLAY] {n / n_1 - n_2 / ALL}	Displays the partition attributes for the partition numbered "n", a range of partitions (n_1 - n_2) or for all partitions.
-DR[OP] .item .item ⋮	Turns off the print flags for the specified items. An item may be the Control Name of any item in Table 18.
EN[D]	Clears the partition definition buffer, closes the partition definition tables to Control access, and returns execution control to the control command level.
{n / n_1 - n_2 / ALL} attribute	Displays the value of the specified attribute for partition number "n", a range of partitions (n_1 - n_2), or for all partitions. "Attribute" must be the name of an attribute listed in Table 18.
{n / n_1 - n_2 / ALL} attribute =number	Sets the value of the specified attribute for partition number "n", a range of partitions (n_1 - n_2), 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 "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. Having operations personnel start STATS as a ghost job each day to generate the snapshot file.
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.

STATS may be run as a batch job, as a ghost job, or on-line. The user must have a privilege level of at least 80. C0 privilege is required if the STATS SET PROC command is to be used. 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 STAT 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 (RET)
STATS D00 HERE
=
```

Table 24. STATS DCB Usage

DCB	Function	On-line Default	Batch Default	Ghost Setting
M:SI ①②	Command input and user responses to interactive statements.	User console	After !STATS card	fid=GHOSTSI. :SYS
M:LO②③	Report output.	User console	Line printer	fid=GHOSTLO. :STATS
M:DO ②	Interactive statements and error messages.	User console	Line printer	fid=GHOSTDO. :STATS
M:OC	Ghost job error messages for operator.	None	None	Operator's Console

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.

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 keyed file with one record for each snapshot interval. The file is opened INOUT, shared. The 7 byte key is of the format year (in binary), day (Julian), hour, minute, second. The records are added chronologically. Each record contains an entry for each computed statistic, the SYSTEM and SWAP histograms (defined later), and a list of the processor names for which CPU time was computed individually. The format of the record is given in Table 25.

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.

Table 25. Format of the SNAPSHOT Record

Word(s)	Contents
0-3	Time and date in TEXT.
4-5	SITEID in TEXT.
6	Number of words in record.
7	Number of display groups.
8	Displacement in words to start of distribution table area.
9	Number of entries in PARAM display.
A	Number of entries in SUMMARY display.
B	Number of entries in CPU display.
C	Number of entries in I/O display.
D	Number of entries in TASK display.

Table 25. Format of the SNAPSHOT Record (cont.)

Word(s)	Contents
E	Number of entries in RAHD display.
F	Number of entries in SCPU display.
10	Number of entries in EVNT display.
11	Number of entries in BATCH display.
12	Number of entries in ONLINE display.
13	Number of entries in USERS display.
14	Number of entries in QUEUE display.
15	Number of entries in RESOURCE display.
16-A0	Values for each item in all of the display groups.
A1-100	Distribution table values for SYSTEM and SWAP distribution.
101-128	Distribution table values for PROC distribution.
129-13C	TEXT name of processors used in BATCH/ ONLINE display.
13D	Contents of SYSVERS (X'2B').
13E-13F	TEXTC name of processor for which extended monitoring is being performed or zero if none is being monitored.
140	Index number of processor being monitored or zero if none is being monitored.

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 13.

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 26. Several of the flags have been described in greater detail in Figures 14 through 29. (These figures are referenced throughout Table 26.) 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. The display is the same as the PARAM display group (see Figure 14) but is not controlled by the PARAM flag.

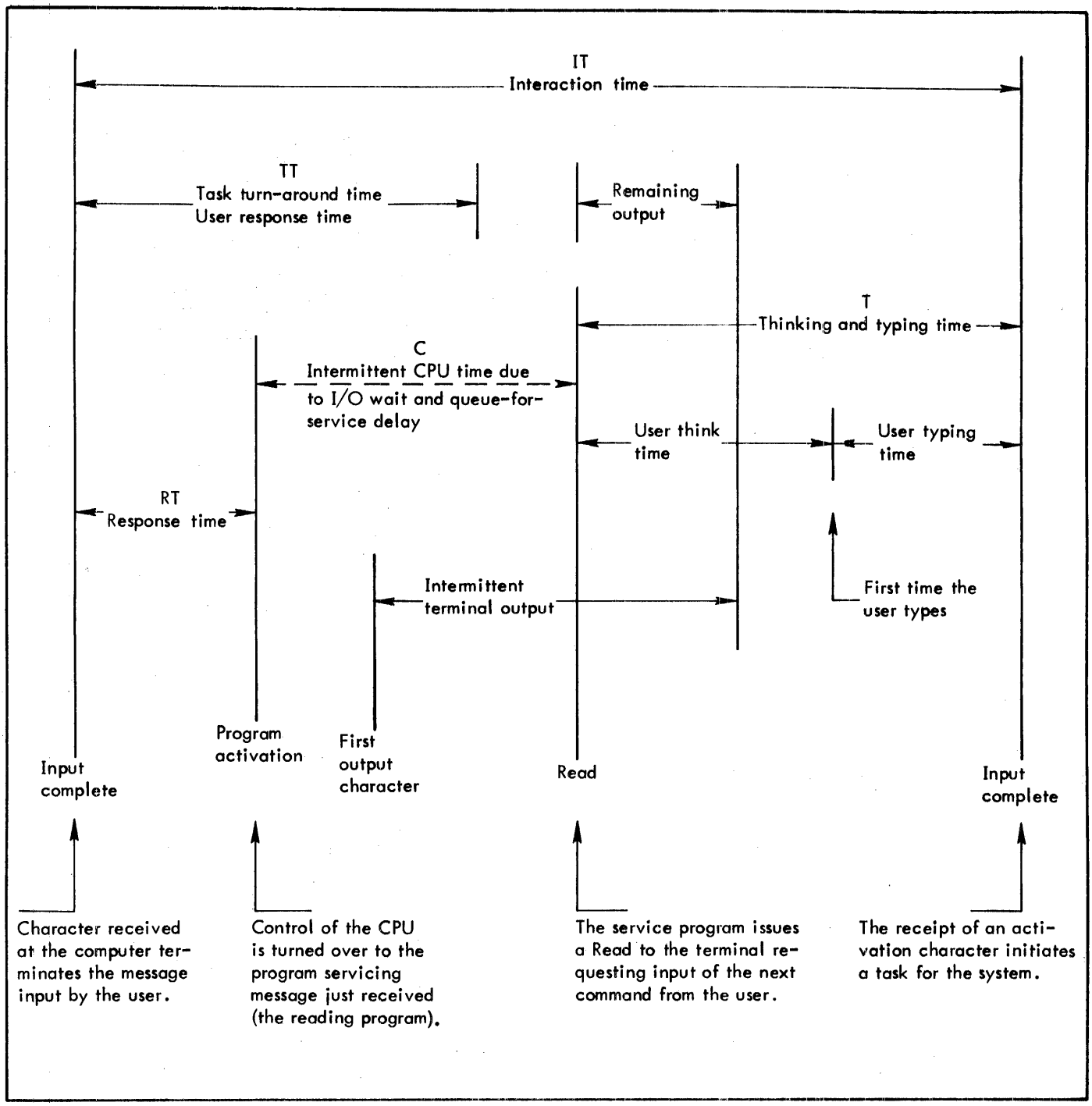


Figure 13. Terminal Interaction Concepts

Table 26. 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 SETPROC command (described later).

Table 26. STATS Flags (cont.)

Item Name	Description
3	A standard display that includes the following display groups (which are described shortly): I/O, TASK, BATCH, ONLINE, USERS, QUEUE, RESOURCE (shown in order of appearance in the display)
4	A standard display that includes the SUMMARY, CPU, and I/O display groups and the single items TIC, UC, OUC, BUC, GUC, and RT90. This is the default display except that the SUMMARY display is omitted.
5	A standard display of the SWAP display group.
6	A standard display that includes the I/O and RAHD groups.
7	A standard display that includes the multiprocessing specific groups, SCPU and EVNT.
ALL	A display which includes the single user items SUMMARY, RAHD, CPU, EVNT, TASK, ONLINE, SCPU, QUEUE, BATCH, SYSTEM, USERS, PROC, RESOURCE, SWAP, I/O (shown in order of appearance in the display)
BATCH	A display that lists the percent of CPU time used by each shared processor for batch programs only. The CPU percentage includes batch execution and service time. (Figure 17 describes this display in detail.)
BUC	A display of the number of batch users currently in the system.
CIT	A display of the total number of interactions received since start-up.
CPU	A display group that includes the percent of CPU time for the monitor and for on-line, batch, and ghost 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 16 describes this display in detail.)
EVNT	A display group of the number of CALs and number of jobs scheduled per minute as measured by the primary CPU and each of the secondary CPUs (SCPU). (Figure 26 describes this display in detail.)
GUC	A display of the number of ghost users currently in the system.
I/O	A display of the I/O rates for the system including symbiont accesses, CALs, and number of interactions per sample minute. (Figure 20 describes this display in detail.)
ONLI[NE]	A display of the percent of CPU time used by on-line users for each of the shared processors. The CPU percentage includes user execution and service time. (Figure 18 describes this display in detail.)
OUC	A display of the number of on-line users currently in the system.
PARA[M]	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 14 describes this display in detail.)

Table 26. STATS Flags (cont.)

Item Name	Description
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 C:PROCN via the SETPROC command. (Figure 24 describes this display in detail.)
QUEU[E]	A display that summarizes the user state queues at the end of each sample interval. (Figure 22 describes this display in detail.)
RAHD	A display group that describes the read ahead and AIR usage in the system. (Figure 27 describes this display in detail.)
REPO[RT]	A nondisplay flag that controls whether or not a report is to be generated while executing the FILE command.
RESO[URCE]	A display group that contains the current usage of certain resources including MPOOLs, CFUs, COCBUFs, and granules (pack, symbiont, and RAD). (Figure 28 describes this display in detail.)
RT90	A display that indicates the 90 percent response time value.
SCPU	A display group that contains the percent of secondary CPU usage for each of the secondary CPUs on the system. These values are broken down into execution and idle time. (Figure 29 describes this display in detail.)
SNAP[SHOT]	A nondisplay flag that controls the generation of snapshot records while executing the DISPLAY command.
SUMM[ARY]	A display that contains key measures of system performance (such as the number of logged users and the measures of response). (Figure 15 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 25 describes this display in detail.)
SYS[TEM]	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 input characters per line. (Figure 23 describes this display in detail.)
TASK	A display of task statistics including interaction time, response time, and task time. (Figure 21 describes this display in detail.)
TIC	A display of the minutes since system start-up.
UC	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.
USER[S]	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 18 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
# CPU ACTIVE                AVE SCPU QUANTUM MSEC
RAHD TIMEOUT SEC            AIR TIMEOUT SEC
```

where

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

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

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 computer-bound users are shared.) The QUAN parameter is set for each partition by the system manager via the Control processor.

Figure 14. Description of the PARAM Display

MSEC ONLINE COMPUTE is the setting of the QUAN control parameter for on-line users. (QUAN is the time-slice by which computer-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 computer-bound queue determine the mix of on-line and batch computing that is done.

CPU ACTIVE is the number of CPUs currently active in the system. This includes both the primary CPU and all secondary CPUs that have been software initialized and available for scheduling.

AVE SCPU QUANTUM MSEC is the average setting of the quantum for all active CPUs. This number is derived from the MAXQ values for each initialized secondary CPU. The MAXQ value is set by the system manager via the Control processor.

RAHD TIMEOUT SEC is the time, in seconds, after which a read ahead operation is aborted. This value is set by the system manager using the name RATO as input to the Control processor.

AIR TIMEOUT SEC is the time in seconds after which an AIR operation is aborted. This value is set by the system manager using the name AIRTO as input to the Control processor.

Figure 14. Description of the PARAM Display (cont.)

The SUMMARY display group is printed in the following format:

```
 SUMMARY              ALL[t]   SNAP[tt]
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.

Figure 15. Description of the SUMMARY Display

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.

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.

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

††SNAP lists statistics for the sample interval only.

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

The CPU display group is printed in the following format:

```
 CPU %          ALL[t]    SNAP[tt]
BATCH EXEC
BATCH SERV
ONLINE EXEC
ONLINE SERV
MONITOR SERV
GHOST EXEC
GHOST 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.

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.

GHOST EXEC is the percent of CPU time spent for ghost users executing in the mapped slave mode.

GHOST SERV is the percent of CPU time spent in the monitor for services required by ghost users. User service for ghost users is always executed in the mapped master mode.

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

SWAP WAIT[ttt] 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[ttt] 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[ttt] 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. Deferred accounting may cause a value of less than 100% to be followed by a value greater than 100% on the next interval. For monoprocessing systems, the total will approximate 100%. For dual processing systems, the total will be as high as 200% with the missing portion representing secondary CPU idle time. See the SCPU display for the idle time.

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

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

[ttt]In a dual processing system, these values are for the primary CPU only.

Figure 16. Description of the CPU Display

The BATCH display group prints in the following format:

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

where the percent of the CPU time for batch users is listed for each of the 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.

*These are the first eight processors that appear on the :SPROCS command of PASS2.

Figure 17. Description of the BATCH Display

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

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

where each of the 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.

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

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

*These are the first eight processors that appear on the :SPROCS command of PASS2.

Figure 18. Description of the ON-LINE Display

The USERS display group prints in the following format:

```
  USERS        #
ANSFORT
AP
APL
BASIC
EDIT
:P00
FORTRAN
:P11
METASYM
PCL
TEXT
IN CORE
```

where each of the listed processors is 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 19. 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
# TRUNCS
# AIR ATTEMPTS
# AIR HITS
SYMBIONTS
IN SWAPS
OUT SWAPS
```

where

SERVICE REQ — is the average number of system CALs executed per minute of sample time on both the primary and all active secondary CPUs.

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.

Figure 20. Description of the I/O Display

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.

TRUNCS is the average number (per minute) of FPOOL buffers that are truncated because they must be shared with other DCBs.

AIR ATTEMPTS is the number of times per minute that an attempt is made to access a file directory.

AIR HITS is the number of times per minute that an attempt to access a file directory results in a non-I/O read. (A non-I/O read is one in which no actual physical I/O operation is performed; i.e., the information is still in a buffer in memory.)

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 outswap.

tALL lists statistics for the period of time since system restart.

ttSNAP lists statistics for the sample interval only.

Figure 20. Description of the I/O Display (cont.)

The TASK display group has the following format:

```
 TASK/INTERVAL    ALL[t]    SNAP[tt]
INTERACT SEC
THINK-TY SEC
TURNARND SEC
COMPLETE SEC
RESPON MSEC
CPU MSEC
```

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.

tALL lists statistics for the period of time since system restart.

ttSNAP lists statistics for the sample interval only.

Figure 21. 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.

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 SRT, SCO, SC1, SC2, SC3, SC4, SC5, SC6, SC7, SC8, and SCU state queues.

COMP BND is the number of users in the SC9 and SC10 state queues. These users are on-line and batch compute bound users.

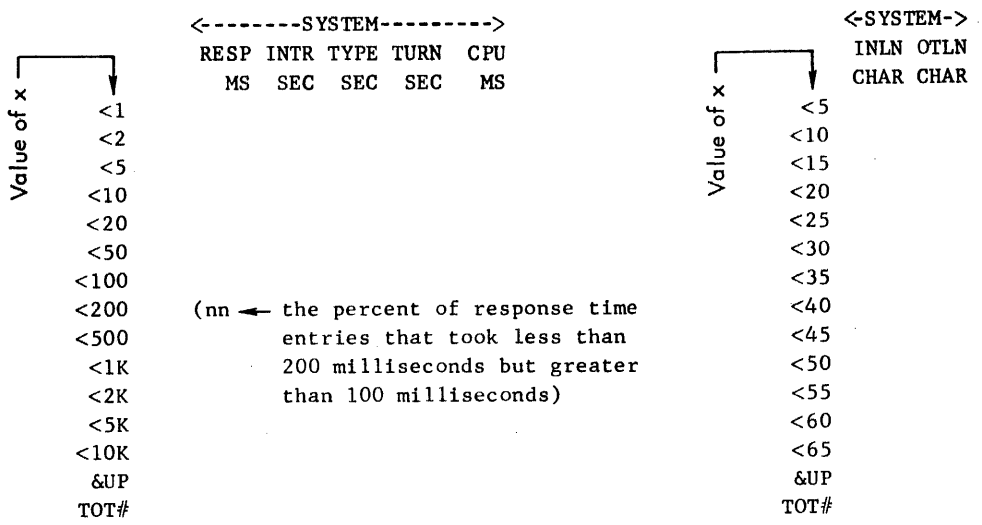
I/O QUEUES is the number of users in the SIOW and SIOMF state queues.

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 22. Description of the QUEUE Display

The SYSTEM display includes seven system histograms in the following format:



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 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 23. Description of the SYSTEM Display

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    #n    name

                <--PROCESSOR-->
                TYPE TURN  CPU
Value of x      SEC  SEC   MS
    ↓          <1
               <2
               <5
              <10
              <20
              <50
             <100
             <200
             <500
              <1K
              <2K
              <5K
             <10K
              &UP
             TOT#
```

```
                <PROCESS->
                 INLN OTLN
Value of x       CHAR CHAR
    ↓           <5
               <10
               <15
               <20
               <25
               <30
               <35
               <40
               <45
               <50
               <55
               <60
               <65
               &UP
              TOT#
```

The heading identifies the processor number as derived from the P:NAME table and also the name of the processor being monitored.

The definitions of the columns are the same as for the SYSTEM display. Note that the RESP and INTR histograms are not included in the PROC display.

Figure 24. Description of the PROC Display

The SWAP display groups includes five histograms of swapper characteristics in the following format:

```
                <--OUT SWAP--->
                               TOT
Value of x      #USERS         MS
    ↓           =0             <1
                =1             <2
                =2             <5
                =3            <10
                =4            <20
                =5            <50
                =6           <100
                =7           <200
                =8           <500
                =9            <1K
               =10            <2K
               =11            <5K
               =12           <10K
                &UP           &UP
               TOT#          TOT#
```

```
<---IN SWAP--->
 JIT REST   TOT
  MS   MS    MS

Value of x
    ↓
   <1
   <2
   <5
  <10
  <20
  <50
 <100
 <200
 <500
  <1K
  <2K
  <5K
 <10K
  &UP
 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.

Figure 25. Description of the SWAP Display

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 not be added together in any way to obtain this histogram.

Figure 25. Description of the SWAP Display (cont.)

The EVNT group display has the following format:

```
EVENT     RATE/MIN    ALL[t]     SNAP[tt]

MASTER CALS
SCPU #1 CALS
SCPU #2 CALS
SCPU #3 CALS
MASTER  SCHED
SCPU #1 SCHED
SCPU #2 SCHED
SCPU #3 SCHED
```

where

MASTER CALS is the average number of system CALs executed by the primary CPU per minute of sample time.

SCPU #n is the average number of system CALs that are initiated on the specified secondary CPU. (This display will only list entries for the number of secondary CPUs generated for the system. STATS alters the format at run time.)

MASTER SCHEDS is the average number of context switches (schedules) per minute on the primary CPU.

SCPU #n SCHEDS is the average number of context switches (schedules) per minute on the specified secondary CPU. (This display will only list entries for the number of secondary CPUs generated for the system. STATS alters the format at run time.)

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

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

Figure 26. Description of the EVNT Display

The RAHD group display has the following format:

```
RAHD - AIR/MIN     ALL[t]     SNAP[tt]

#AIR ATTEMPTS
#AIR HITS
#AIR TIMEOUTS
#RAHD ATTEMPTS
#RAHD STARTS
#RAHD USED
#RAHD I/O WAIT
#RAHD NOT USED
#RAHD TIMEOUTS
```

where

#AIR ATTEMPTS is the number of times per minute that an attempt is made to access a file directory.

#AIR HITS is the number of times per minute that an attempt is made to access a file directory which results in a non-I/O read.

#AIR TIMEOUTS is the number of times per minute that file directory granules in core are released because they were not accessed within the timeout interval.

#RAHD ATTEMPTS is the number of times per minute that an attempt was made to initiate a read ahead operation.

#RAHD STARTS is the number of times per minute that a read ahead operation is actually started. (The number of RAHD ATTEMPTS should always be larger than or equal to the number of RAHD STARTS.)

#RAHD USED is the number of times per minute that those read ahead starts were actually used.

#RAHD I/O WAIT is the number of times per minute that when attempting to use a read ahead granule it was found that the I/O had not yet completed. (The number of I/O WAITS should be less than or equal to the number USED.)

#RAHD NOT USED is the number of times per minute that a read ahead operation was started but was discarded because it was not needed.

#RAHD TIMEOUTS is the number of times per minute that a read ahead buffer was released because it was not used within the timeout interval. (The number of USED + NOT USED + TIMEOUTS should be less than or equal to the number of STARTS.)

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

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

Figure 27. Description of the RAHD Display

The RESOURCE Group display has the following format

```
RESOURCES IN USE      #
MPOOLS
COCBUFS
IOQ ENTRIES
CFUS
GRAN PACK
GRAN SYMB
GRAN RAD
```

where the numbers for each of the entries are evaluated only at the end of each sample interval and do not represent an average count for the interval.

Figure 28. Description of the RESOURCE Display

The SCPU group display has the following format:

```
SCPU USE %     ALL[t]     SNAP[tt]

SCPU #1 EXEC
SCPU #1 IDLE
SCPU #1 TOT
SCPU #2 EXEC
SCPU #2 IDLE
SCPU #2 TOT
SCPU #3 EXEC
SCPU #3 IDLE
SCPU #3 TOT
```

where

SCPU #n EXEC represents the percent of time the specified secondary CPU is executing in the slave mode.

SCPU #n IDLE represents the percent of time the specified secondary CPU is inactive.

SCPU #n TOT represents the total time for the specified secondary CPU.

If all three entries for a given secondary CPU are zero, then the CPU is not running (i.e., it has not been initialized). If the total value is less than 100%, the fraction that is missing represents the time spent in monitor services for a user.

This display will only list entries for the number of secondary CPUs generated for the system. STATS alters the format at run time.

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

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

Figure 29. Description of the SCPU Display

STATS COMMANDS

The STATS commands will be described in the following order:

HELP	FILE
ADD	BREAK (i.e., BRK)
DROP	PROCEED
BUILD	TIME
LIST	SETPROC
DISPLAY	END
CONTROL!	

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 30. The format of the command is

H[ELP]

```
      DESCRIPTION OF STATS COMMANDS
A(DD)          -  ADD PRINT FLAGS
B 0            -  RESET PRINT FLAGS
B(UILD) 1      -  SET FLAG SYSTEM DISTRIBUTION
B 2            -  SET FLAGS SYSTEM & PROCESSOR
                  DISPLAYS
B 3            -  SET FLAGS 7 GROUPS
B 4            -  SET FLAGS CPU, I/O & SINGLE ITEMS
B 5            -  SET FLAG SWAP DISTRIBUTION
B 6            -  SET FLAGS I/O AND RAHD/AIR
B 7            -  SET FLAGS SCPU AND EVNT
C(ONTROL!)     -  LIST SOME OF CONTROL PARAMS
DI(SPLAY)      -  REPORT & UPDATE SNAPSHOT FILE
D(ROP)         -  RESET PRINT FLAGS
E(ND)          -  EXIT STATS
F(ILE)         -  UPDATE SNAPSHOT FILE & REPORT
L(IST)         -  LIST CURRENT PRINT FLAG STATUS
T(IME)         -  PRINT CURRENT TIME
S(ETPROC)      -  SET PROCESSOR NUMBER FOR
                  MONITORING
      DCB USAGE
M:SI           -  INTERACTIVE INPUT COMMANDS
M:LO           -  REPORTS
M:DO           -  INTERACTIVE OUTPUT
F:3            -  SNAPSHOT FILE
```

Figure 30. 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 in Table 26. When the STATS processor is entered, CPU, I/O, and the single items TIC, UC, OUC, BUC, GUC, and RT90 are on by default. The item names are entered following prompts. (Only the first four characters are required.) A null response (or END) to a prompt terminates the list of items. The format of the command is

A[DD]

Example:

```
=A (RET)
 ENTER ITEMS TO BE ADDED
>RAHD (RET)
>RESO (RET)
>SCPU (RET)
>EVNT (RET)
> (RET)
=
```

The ADD command enables the user to select a group of parameters to be displayed whenever the DISPLAY or FILE 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 26. The item names are entered following prompts. (Only the first four characters are required.) A null response (or END) to a prompt terminates the list of items. The format of the command is

D[ROP]

Example:

```
=D (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 or turns off all 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 26.) If n is the digit 0, all print flags are turned off. If more than one digit is specified, the first digit is used.

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

L[IST]

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.

```
-L (RET)
 FLAG        -        STATUS
PARAM        -          OFF
CIT          -          OFF
TIC          -   ON
UC           -   ON
OUC          -   ON
BUC          -   ON
GUC          -   ON
RT90         -   ON
SUMMARY      -          OFF
CPU          -   ON
I/O          -   ON
TASK         -          OFF
RAHD         -          OFF
SCPU         -          OFF
EVNT         -          OFF
BATCH        -          OFF
ONLINE       -          OFF
USERS        -          OFF
QUEUE        -          OFF
RESOURCE     -          OFF
SYSTEM       -          OFF
PROC         -          OFF
SWAP         -          OFF
SNAPSHOT     -          OFF
REPORT       -          OFF
-
```

2. In this example, the flag for standard display 7 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.

```
-B 7 (RET)
-L (RET)
 FLAG        -        STATUS
PARAM        -          OFF
CIT          -          OFF
TIC          -   ON
UC           -   ON
DUC          -   ON
BUC          -   ON
GUC          -   ON
RT90         -   ON
SUMMARY      -          OFF
CPU          -   ON
I/O          -   ON
TASK         -          OFF
RAHD         -          OFF
SCPU         -   ON
EVNT         -   ON
BATCH        -          OFF
ONLINE       -          OFF
USERS        -          OFF
QUEUE        -          OFF
RESOURCE     -          OFF
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 26) 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

DI[SPLAY]

Example:

```
-DI (RET)
 ENTER INTERVAL IN MINUTES
≥5 (RET)
 ENTER # OF INTERVALS
≥4 (RET)
```

If the output device cannot finish printing on 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 file. A report may be generated simultaneously by turning the REPORT flag and the desired print flags on. The format of the FILE command is

F[ILE]

Example:

```
=F ↵
 ENTER INTERVAL IN MINUTES
≥5 ↵
 ENTER # OF INTERVALS
≥4
 SNAPSHOT FILES OPENED
..SLEEPING
 RECORD # 1  14:35 JUL 21, '75
..SLEEPING
 RECORD # 2  14:40 JUL 21, '75
..SLEEPING
 RECORD # 3  14:50 JUL 21, '75
..SLEEPING
 RECORD # 4  14:55 JUL 21, '75
..SLEEPING
 RECORD # 5  15:00 JUL 21, '75
 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

P[ROCEED]

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

T[IME]

Example:

```
=T ↵
14:30 JUL 21, '75
```

SETPROC This command enables the user to specify a processor for extended monitoring. In order to use this command, STATS must be run in an account with a C0 privilege. The format of the command is

S[ETPROC]

When the command is given, STATS prompt for the processor name with the message

ENTER PROCESSOR NAME

When the name is entered, STATS locates the name in the shared processor table and stores its index number into the C:PROCN table in the monitor root. A subsequent PROC display will give the statistics gathered on this processor.

Example:

```
=S ↵
 ENTER PROCESSOR NAME
≥EDIT ↵
 PROCESSOR NOW BEING MONITORED
 PROCESSOR #29 EDIT
=
```

To reset the processor monitoring cell so that no processor is being monitored, the following technique is used prior to exiting STATS:

```
=S ↵
 ENTER PROCESSOR NAME
≥0 ↵
 PROCESSOR MONITORING STOPPED
=END ↵
```

The following example illustrates what happens when a processor is already being monitored at the time SETPROC is used:

```
=S ↵
 ENTER PROCESSOR NAME
≥LINK ↵
 REPLACED A PROCESSOR BEING MONITORED
 PROCESSOR NOW BEING MONITORED
 PROCESSOR #23 LINK
```

If the processor name specified is not found in the P:NAME table, then a message is output and control returns to input another name as is illustrated in the following example:

```
=S ↵
 ENTER PROCESSOR NAME
≥GRUNCH ↵
 PROCESSOR NOT SHARED
 ENTER PROCESSOR NAME
≥
```

END This command causes an exit to TEL. The format of the command is

END

Example:

```
-END ↵
!
```

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 27.

STATS COMMAND SUMMARY

Table 28 contains a summary of commands. Table 29 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:10 LP ↵
!STATS ↵
STATS DOO HERE
-A ↵
 ENTER ITEMS TO BE ADDED
≥ALL ↵
≥END ↵
-DI ↵
 ENTER INTERVAL IN MINUTES
≥2 ↵
 ENTER # OF INTERVALS
≥10 ↵
 -END ↵
 !PRINT ↵
```

The !PRINT command releases the listing to the printer.

Example 2.

Create snapshot files on-line.

```
!STATS ↵
STATS DOO HERE
-F ↵
 ENTER INTERVAL IN MINUTES
≥5 ↵
 ENTER # OF INTERVALS
≥2 ↵
 SNAPSHOT FILES OPENED
 RECORD # 1, 14:10 JUL 21, '75
 RECORD # 2, 14:15 JUL 21, '75
 SNAPSHOT FILE COMPLETED
-E ↵
```

Example 3.

Run STATS as a ghost job. Every 15 minutes, snapshot records are added to the SNAPSHOT.:STATS file. The job will "sleep" between times.

The job is initiated by the operator via the key-in

!GJOB STATS ⓃⓁ

The file GHOSTSI contains the following instructions to STATS:

```
*        STATS SLEEPING GHOST INPUT FILE TO CREATE SHAPSHOT 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 - KEYED SNAPSHOT FILE
*                GHOSTDO.:STATS - INTERACTIVE OUTPUT
*                GHOSTLO.:STATS - REPORT OUTPUT
*        SNAPSHOT FILE MAY BE SAVED WEEKLY TO SAVE FILE SPACE
F
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 file is opened. At the conclusion, another message is typed on the operator's console to verify that the snapshot file is closed.

Table 27. STATS Error Messages

Message	Description
BUILD LIST ERROR	The BUILD list number is missing or is not in the range 0-7.
CANNOT MAINTAIN INTERVAL	During operation of a DISPLAY or FILE 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 attempted to create 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 attempted to create a disk file 'GHOSTLO' that is in error. The error is due either to lack of file space or to inerference from another user. (Printed on operator's console.)
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.)

Table 27. STATS Error Messages (cont.)

Message	Description
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.)
MONITOR ACCESS REQUIRES PRIV = 80	The user account must be reauthorized to allow the user to read monitor performance data.
NO SUCH NAME	An invalid name was used when entering items for an ADD or DROP command. Only the item names in Table 26 are valid. The standard display numbers are not valid for the DROP command.
NOT ENOUGH CORE AUTHORIZED	The number of pages allocated in the user account is not sufficient since STATS only maps onto those pages in which the required data resides and obtains the corresponding virtual pages, the processor and its acquired pages should be executable in 16K of core.
PROBLEM GETTING PRIVATE PAGE	In a multiprocessing system, an error has been encountered in attempting to map onto the private data page of the secondary CPUs. This may be due to a system or program error and, generally, should not occur.
PROCESSOR #n name	Extended monitoring is being performed for the processor specified by 'name' which has the index number n.
PROCESSOR MONITORING STOPPED	STATS is no longer performing extended monitoring for a processor.
PROCESSOR NOW BEING MONITORED	Extended monitoring is being performed for the processor named in the message which follows this message.
REPLACED A PROCESSOR BEING MONITORED	Extended monitoring was already being performed for a processor when the SETPROC command was issued.
SETPROC REQUIRES C0 PRIVILEGE	An attempt has been made to establish extended monitoring of a processor when running in an account that has less than C0 privilege.
STATS LOADED WITH WRONG MONSTK	STATS has been loaded with a MONSTK that is different from the system MONSTK. The processor must be reloaded.
STATS SNAPSHOT FILE ERROR	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.
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.

Table 28. STATS Command Summary

Command	Description
-A[DD] 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 26.
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.
B[UILD] n	Turns on the print flags for the standard display specified by n where n is a number in the range 0-7. See Table 26 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.
DI[SPLAY]	Displays those items (from the group of items listed in Table 26) that have their print flags on. A snapshot file may be generated during DISPLAY command execution if the SNAPSHOT flag is on.
-D[ROP] 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 26.
E[ND]	Causes an exit to TEL.
END	Terminates a DROP or ADD command list.
F[ILE]	Creates snapshot records for the SNAPSHOT file. An on-line report may be generated simultaneously by turning the REPORT flag and the desired print flags on.
H[ELP]	Briefly describes STATS commands and DCB usage.
L[IST]	Lists the STATS flags (other than standard display numbers) and their current status (on or off).
P[ROCEED]	Continues STATS operation from the point at which the processor was interrupted by the BREAK key.
S[ETPROC]	Establishes the processor for extended monitoring.
TIME	Lists the time and date.

Table 29. 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 or FILE 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 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 or FILE command.
ENTER PROCESSOR NAME	Enter the name of a shared processor to be monitored. The name must be less than 8 characters in length and must be the name by which the processor is identified in the P:NAME table.

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 SNAPSHOTS files created by STATS. The output listings are generally large and therefore should be output to a file or on the line printer. The DCBs used by Summary are listed in Table 30.

The Summary processor allows the user to

1. Specify a filter to remove undesired snapshots from the sample for subsequent reports.
2. Request a chronological listing of filtered snapshot data for one or more display groups.
3. Request filtered, sorted, ordered, and averaged listings of snapshot data for one or more display groups.
4. Request means, minimums, maximums, and standard deviations for all display groups computed using the snapshots which pass the 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.

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 or END 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 31. 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 32. The format of the command is

```
LIST
```

Table 30. 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
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.

```
SUMMARY IS A POST PROCESSOR FOR THE STATS PROCESSOR
REQUIRES SNAPSHOT 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
 RECOMMENDED STANDARD PROCEDURE FOLLOWS
YES       (SPECIFY FILTER ?)
38,0,10        (LESS THAN 10% CPU IDLE)
END       (TERMINATES FILTER LIST)
YES       (CHRONOLOGICAL SUMMARIES ?)
NO        (AVERAGES ONLY ?)
2         (CPU)
6         (SCPU)
NO        (MORE GROUPS ?)
YES       (CORRELATION ANALYSIS ?)
          (DEFAULT)
          (DEFAULT)
SPECIFY FILTER      ?
*STOP* 0
```

Figure 31. 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 # SEC TERM BLK
 8 # SEC TERM UNLK
 9 QMIN          MSEC
10 SQUAN         MSEC
11 AVE BATCH QUAN
12 MSEC ONLINE QUAN
13 CPU ACTIVE
14 AVE SCPU QUAN MS
15 RAHD TMOUT SEC
16 AIR TMOUT  SEC

   1 - SUMMARY
17 BATCH STREAMS
18 ONLINE USERS
19 % CPU/BATCH USER
20 % CPU/ONLINE USE
21 BATCH EXEC/SERV
22 ONLINE EXEC/USER
23 CPU MSEC PER I/O
24 ONLINE TIME MIX
25 ONLINE INTENSITY
26 ONLINE TASKS/MIN
27 % INTERACTIVE
28 90% RESPONSE  MS
29 TURNAROUND   SEC
30 ETMF

   2 - CPU
31 BATCH EXEC
32 BATCH SERV
33 ONLINE EXEC
34 ONLINE SERV
35 MONITOR SERV
36 GHOST EXEC
37 GHOST SERV
38 IDLE
39 SWAP WAIT
40 I/O WAIT
41 I/O & SWAP WAIT
42 TOTAL

   3 - I/O
43 SERVICE REQUESTS
44 INTERACTIONS
45 CHAR IN
46 CHAR OUT
47 TERM WRITES
48 I/O ACCESSES
49 # TRUNCS
50 AIR ATTEMPTS
51 AIR HITS
52 SYMBIONT
53 IN SWAPS
54 OUT SWAPS

   4 - TASK
55 INTERACT
56 THINK-TY
57 TURNARND
58 COMPLETE
59 RESPONSE     MSEC
60 CPU          MSEC

   5 - RAHD
61 AIR ATTEMPTS
62 AIR HITS
63 AIR TIMEOUTS
64 RAHD ATTEMPTS
65 RAHD STARTS
66 RAHD USED
67 RAHD I/O WAIT
68 RAHD NOT USED
69 RAHD TIMEOUTS
```

Figure 32. LIST Command Listing

```
   6 - SCPU
70 SCPU 1 EXEC
71 SCPU 1 IDLE
72 SCPU 1 TOT

   7 - EVNT
73 MASTER CALS
74 SCPU 1 CALS
75 MASTER SCHED
76 SCPU 1 SCHED

   8 - BATCH
77 CCI
78 LOGON
79 LINK
80 DELTA
81 :P00
82 :P11
83 DRSP
84 RUNNER
85 EDIT
86 PCL
87 USER
88 SHARED

   9 - ONLINE
89 CCI
90 LOGON
91 LINK
92 DELTA
93 :P00
94 :P11
95 DRSP
96 RUNNER
97 EDIT
98 PCL
99 USER
100 SHARED

  10 - USERS
101 ANSFORT
102 AP
103 APL
104 BASIC
105 EDIT
106 FDP
107 FORTRAN
108 FORTLIB
109 METASYM
110 PCL
111 TEXT
112 IN CORE

  11 - QUEUE
113 LOGGED
114 LOGGING
115 BATCH
116 GHOST
117 TERM IN
118 TERM OUT
119 COMPUTE
120 COMP BND
121 I/O
122 SLEEP

  12 - RESOURCE
123 MPOOLS
124 COCBUFS
125 IOQ ENTRIES
126 CFUS
127 GRAN PACK
128 GRAN SYMD
129 GRAN RAD
```

Figure 32. 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 response to these questions and statements define the tasks required of Summary. There are three major questions asked by Summary. They are asked in the following order:

1. SPECIFY FILTER?
2. CHRONOLOGICAL SUMMARIES?
3. CORRELATION ANALYSIS?

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 33 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

FILTER SPECIFICATION

It is extremely useful to be able to reject snapshots which fail to meet certain criteria. The filter specification 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 filters by the question 'SPECIFY FILTERS?'. His response may be one of the following:

YES which indicates that he would like to specify 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 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 filter. Of course, any limits that are appropriate for the needs of the particular installation may be chosen.

CHRONOLOGICAL SUMMARIES

A chronological report of system performance statistics may include from one to fourteen statistical groups of data. The statistical groups are listed in Table 31, along with the STATS flag associated with each statistical group.

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.

Following a response of YES or ALL, Summary then types the statement

AVERAGES ONLY?

The response to this statement may be

YES which requests that only the average values for each statistical group be printed.

NO which requests that all values be printed.

{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 to the 'CHRONOLOGICAL SUMMARIES' query was YES, Summary then types the statement

ENTER ALL, OR GROUP #0-13

The user's response to this statement may be

ALL which requests that all statistical groups be displayed.

n which specifies the number (0-13) of a statistical group to be displayed. If the number is outside the range 0-13, the statement is repeated.

{X / END} which stops the Summary processor and returns control to TEL.

other which causes the next major question 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 #0-13' statement.

0-13 which also requests that all statistical groups be displayed.

other which causes the next major question 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 SNAP SHOT 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.

Table 31. Summary Statistical Groups

Group No.	STATS Flag	Description
0	PARAM	Selected control parameters.
1	SUMMARY	Performance statistics.
2	CPU	CPU utilization: execution, service, idle, and swap wait.
3	I/O	I/O rates.
4	TASK	Terminal interaction statistics.
5	RAHD	Read ahead and AIR usage.
6	SCPU	Secondary CPU utilization.
7	EVNT	CAL and execution scheduling activities.
8	BATCH	CPU time used by batch shared processors.
9	ONLINE	CPU time used by on-line shared processors.
10	USERS	Shared processor utilization.
11	QUEUE	State queue statistics.
12	RESOURCE	System resource utilization summary.
13	HISTOGRAM	System histograms for system and swap statistics.

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 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 0, 1000000000 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 34 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 35 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 36.)

(This page is intentionally blank.)

```
OVERALL SUMMARY OF   14 SNAPSHOTS SELECTED FROM   17
INTENSITY RANGE FROM        0. TO 000000000.
FILTER RESTRICTIONS FOR SAMPLE
   #                          LOW       HIGH      # LOW     # HIGH
  34 IDLE                     .0        10.0        .0        3.0
    0 = PARAM
   #                        MEAN       MIN       MAX     STD.DEV.
   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
   #                        MEAN       MIN       MAX     STD.DEV.
  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
   #                        MEAN       MIN       MAX     STD.DEV.
  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
   #                        MEAN       MIN       MAX     STD.DEV.
  39 LINK
  40 DELTA
  41 :P00
  42 :P11
  43 LDEV
  44 EDIT
  45 PCL
  46 BASIC
  47 METASYM
  48 LOADER
  49 USER
  50 SHARED
```

Figure 34. 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 35. A Section of the Covariance Matrix

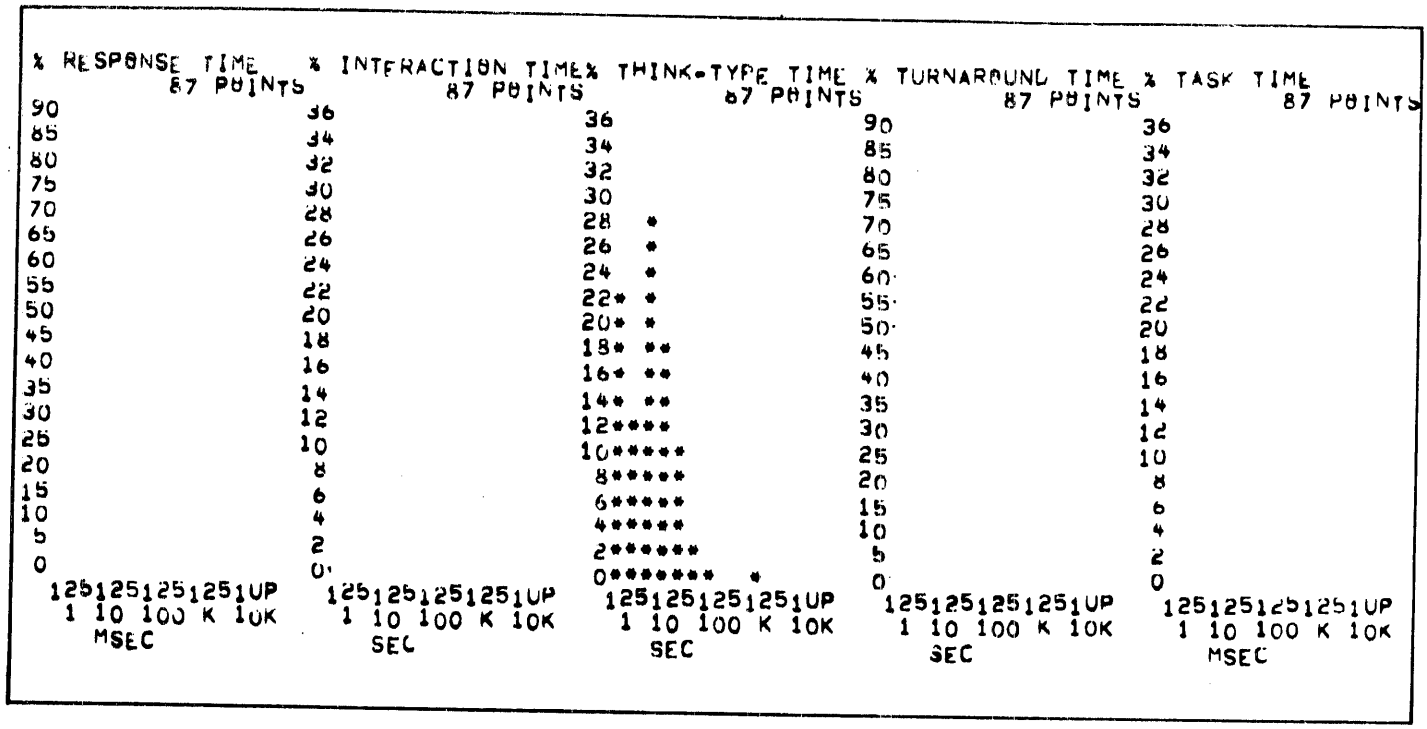


Figure 36. Task and Swap Statistics

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
 SPECIFY FILTER?
=NO⏎
 CHRONOLOGICAL SUMMARIES?
=ALL⏎
 AVERAGES ONLY?
=NO⏎
 CORRELATION ANALYSIS?
=YES⏎
 ENTER CORRELATION THRESHOLD (0.0 TO .99)
≥.8⏎
 THRESHOLD = .80
 ENTER INTENSITY RANGE. (XLOW,XHIGH)
≥0,1000.⏎
 INTENSITY RANGE FROM 0. TO 1000.
 100 SNAPSHOTS WERE SELECTED FROM 100
 *STOP* 0

!PRINT⏎
```

2. Set filter limits to restrict the control parameters within limits and produce a filtered, averaged summary of all statistical groups.

```
!SET M:LO LP⏎

!SUMMARY⏎
 SUMMARY HERE
 SPECIFY FILTER?
=YES⏎
 ENTER ITEM#, LOW LIMIT, HIGH LIMIT
≥1,13.,15.⏎
   1 HOUR:MINUTES          1.30      15.0
≥ ⏎
 CHRONOLOGICAL SUMMARIES?
=ALL⏎
 AVERAGES ONLY?
=YES⏎
 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 #0-13' interaction.

```
!SET M:LO LP

!SUMMARY ↵
 SUMMARY HERE
 SPECIFY FILTER?
=NO↵
 CHRONOLOGICAL SUMMARIES?
=YES↵
 AVERAGES ONLY?
=NO ↵
 ENTER ALL, OR GROUP #0-13
≥1 ↵
 ANOTHER GROUP?
≥2 ↵
 ANOTHER GROUP?
≥YES
 ENTER ALL, OR GROUP #0-13
≥5 ↵
 ANOTHER GROUP?
≥  ↵
 CORRELATION ANALYSIS?
=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
NO        (SPECIFY FILTER?)
ALL       (CHRONOLOGICAL SUMMARIES?)
NO        (AVERAGES ONLY?)
YES       (CORRELATION ANALYSIS?)
.3        (ENTER CORRELATION THRESHOLD.
               (0.0 TO .99))
200,2200  (ENTER INTENSITY RANGE.
               (XLOW, XHIGH))
!EOD
```

5. Get overall statistical summary without filtering. Summary is run as a batch job.

```
!SUMMARY
NO        (SPECIFY FILTER?)
NO        (CHRONOLOGICAL SUMMARIES?)
YES       (CORRELATION ANALYSIS?)
blank     (ENTER CORRELATION THRESHOLD.
               (0.0 TO .99))
blank     (ENTER INTENSITY RANGE.
               (XLOW,XHIGH))
!EOD
```

SUMMARY ERROR MESSAGES

If Summary tries to open SNAPSHOT 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 32 and 33 respectively.

Table 32. 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 # 0-13' statement. {ALL / 0-13} -requests that all statistical groups be displayed. other -leads to the next major interactive question.
AVERAGES ONLY?	NO -requests that average values for all statistical groups be displayed after filtering. YES -requests that average values for one or more statistical groups be displayed after filtering. 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 Questions (cont.)

Questions	User Response
CHRONOLOGICAL SUMMARIES?	ALL - requests that all statistical groups be displayed chronologically. YES -requests that one or more statistical groups be displayed chronologically. 'ENTER ALL, OR GROUP #0-13' 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.
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.
SPECIFY 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 33. Summary Interactive Statements

Statement	User Response
ENTER ALL, OR GROUP #0-13	ALL -has the same meaning as an ALL answer to the question asked just before this statement. n -specifies the number (0-13) of a statistical group to be displayed. If the number is outside the range 0-13, 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 33. 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 1000000000 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. SYSTEM PERIPHERAL CONTROL

SYSCON

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 cannot be partitioned.
  - Teletypes
  - COCs
  - RAF
  - Public disk pack spindles
- A partitioned resource may be returned to the system.
- The status of 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.

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 ®
SYSCON HERE
>
.
.
.
```

SYSCON COMMAND SYNTAX

Many of the SYSCON commands contain the following parameter as part of the command format:

[yy] value

The description of this parameter is quite detailed. To avoid repeating the description several times, it will be given here and references will be made back to this section in the command descriptions.

The yy portion of the parameter specifies the device type (e.g., LP, 9T). The brackets indicate that the yy portion is unnecessary and optional. However, if the yy portion is specified, it is verified by SYSCON. Also, note that there must be no space between yy and value.

The format of the value portion is dependent on the CPU being used.

For Sigma 6, 7, and 9 systems, value must be in the format

ndd

where

n represents an IOP address and is specified as a letter. See Table B-2 in Appendix B.

dd specifies the device number. See Table B-3.

For Xerox 560 systems, value may take one of two formats. The first format is

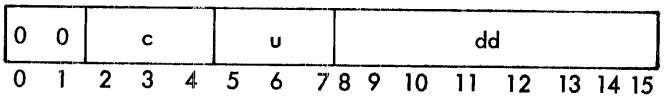
ndd

where

n represents a cluster number and a unit number and is specified as a letter. See Table B-4.

dd specifies the device number. See Table B-3.

The second format consists of four hexadecimal digits which represent a hardware address in the format shown below.



where

c specifies the cluster number.

u specifies the unit number.

dd specifies the device number.

SYSCON COMMANDS

DISPLAY The DISPLAY command displays the status of resources. The format of the command is:

```
                [[yy] value
                yy
DI[SPLAY]       CONT[,[yy] value]
                PART
                ALL
```

where

[yy] value requests the status for the specified device. See 'SYSCON Command Syntax' above.

yy requests the status for all devices of type yy.

CONT[,[yy] value] requests the status of all controllers within the system if the [yy] value portion is omitted, or requests the status of just the controller for the device specified by [yy] value. (See 'SYSCON Command Syntax' above.)

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
--------------------------------------------------
    yyndd               {PART                }
                        {IDLE  } [NON-PART]  }
                        {uid   }             }
                        {blank }             }

    CONT,yyndd          {PART    } [PRIM]
                        {UP      } [ALT ]
                        {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.

PRIM indicates that the controller is the primary path for pooled devices.

ALT indicates that the controller is the alternate path for pooled devices.

**** indicates that the controller resides in a dual access channel but is not pooled.

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.)

The partitioning of an allocatable resource (e.g., tape), is not allowed when the total number of resources currently allocated is equal to the total number of resources available for that device type. This also applies to a partition controller request where the request results in the partitioning of devices in the controller and the total number of resources currently allocated is equal to the total number of resources available for that device type. However, one or more of the devices within the controller may end up partitioned.

When the device is a symbiont, the lsyyndd, L key-in (for input symbiont) or lsyyndd, Q key-in (for output symbiont) is simulated. 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:

```
                {[yy] value      }
PA[RTITION]     {CONT,[yy] value }
```

where

[yy] value specifies the address of the device that is to be partitioned. See "SYSCON Command Syntax' above.

CONT,[yy] value specifies the name of a controller that is to be partitioned. See 'SYSCON Command Syntax' above. When a controller is partitioned, all of its devices are also partitioned unless the controller is dual access. When the controller is dual access, only the path specified by value is partitioned unless the other path to the device is already partitioned or doesn't exist (i.e., single access within dual channel). In the latter case, all of the controller's devices are partitioned. If any of its devices cannot be partitioned for any reason, then the controller (and its devices) will not be partitioned. However, some of the devices may be partitioned.

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] value / CONT,[yy] value}

where

[yy] value specifies the name of a device that is to be returned to the system. See 'SYSCON Command Syntax' above.

CONT,[yy] value specifies the name of a controller that is to be returned. See 'SYSCON Command Syntax' above. When it is requested that a controller be returned to the system and one or more devices within the controller are busy with on-line diagnostics, the controller will not be returned.

MODNUM The MODNUM command is used to generate, update, or display the contents of the file M:MODNUM. M:MODNUM is a keyed file in the format of a load module 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. This file is also used by the boot-time reconfiguration process for boot-time control command verification.

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 ↵
!SYSCON ↵
```

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:

MO[DNUM] {NEW / UPDATE / DISP}

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 37.

```
**********************************************
    DEVICE              CONTROLLER
----------------------------------------------
      7120                 7120
       :                    :
       :                    :
      7242                 7240
       :                    :
       :                    :
**********************************************
```

Figure 37. 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#

D {device model# / controller model#}

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

LIST This command causes SYSCON to send output to the M:LO device when SYSCON is being run in the on-line or ghost mode. (The LIST command is ignored in the batch mode.) In the on-line mode, a ISET M:LO LP command must be given prior to entering SYSCON. Otherwise, M:LO is defaulted to M:UC. The format of the command is

LIST

NOLIST This command causes SYSCON to output to the M:UC device in the on-line or ghost mode (which implies the user's terminal or the operator's console, respectively). The NOLIST command is ignored in the batch mode. The format of the command is

NOLIST

END The END command terminates SYSCON. The format of the command is

END

SAMPLE SYSCON SESSION

Figure 38 is a sample on-line SYSCON session in which a line printer and a controller with four 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.

```
1SYSCON (RET)
SYSCON HERE


≥PA LPAOF (RET)
SYMBIONT LPAOF TERMINATED
LPAOF PARTITIONED


≥PA CONT,9TA80 (RET)
9TA80 PARTITIONED
9TA81 PARTITIONED
9TA82 PARTITIONED
9TA83 PARTITIONED
CONT. PARTITIONED

≥DI PART (RET)


**********************************
  RESOURCE                STATUS

  ------------------------------


LPAOF                   PART
9TA80                   PART
9TA81                   PART
9TA82                   PART
9TA83                   PART
CONT,9TA80              PART
**********************************

 ≥END (RET)
```

Figure 38. 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.

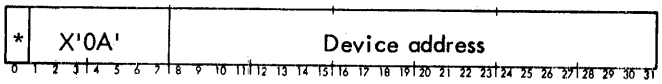
device address has the format described for 'value' in the section 'SYSCON Command Syntax' above.

Calls generated by the M:DPART procedure have the form

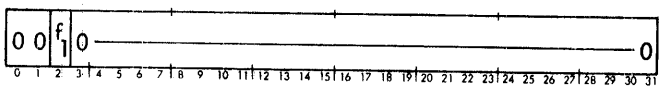
CAL1,6 fpt

where fpt points to word 0 of the FPT shown below.

word 0



word 1



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 resource cannot be partitioned.

CC2=0 device address is valid; partitioning completed.

CC2=1 unknown device address; partitioning request ignored.

CC4=1 partitioning of a controller resulted in only a controller path being partitioned.

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.

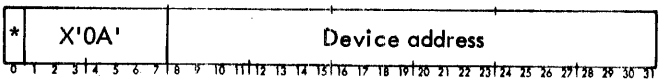
device address has the format described for 'value' in the section 'SYSCON Command Syntax' above.

Calls generated by the M:DRET procedure have the form

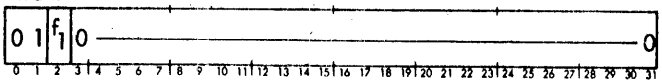
CAL1,6 fpt

where fpt point to word 0 of the fpt shown below.

word 0



word 1



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.

CC2=0 device address is valid; return completed.

CC2=1 unknown device address; return request ignored.

CC4=1 return of a controller resulted in only a controller path being returned.

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 34) 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 35) 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 34. 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.
*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, yyndd ALREADY PARTITIONED	The specified resource is already partitioned. The command is ignored.
CANNOT PARTITION, yyndd 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, yyndd 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.
CONTINUATION ILLEGAL	A semicolon was encountered but continuation commands are not accepted.
CONTROLLER HAS BUSY DEVICE	A controller return request cannot be honored since one of the controller's devices is busy with on-line diagnostics.
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.
yyndd 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.
yyndd 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.

Table 34. SYSCON Messages for the User's Terminal (cont.)

Message	Description
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.
PARTITIONED yyndd 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.
SOME DEV. IN CONTROLLER MAY BE PARTITIONED	A controller partition request was made. SYSCON started partitioning devices and then encountered a nonpartitionable device. Therefore, the controller is not partitioned but some of its devices may have been partitioned.
SYMBIONT yyndd TERMINATED	The specified symbiont device or the controller associated with the specified symbiont device has been partitioned.
SYSCON HERE	SYSCON has been entered.
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 35. 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.

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.

FIX

The Fix processor performs the following functions for the file management system:

- HGP reconstruction.
- Consistency checking.
- Repair of directories and files.
- Deletion of account and file names from directories without release of granules.

The processor is an integral part of booting a system and of a recovery. It may be run on-line, as a ghost, or in batch.

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 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. 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 if 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.

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. Proper operation of GAC is dependent upon having unique account fields for each user. (The account field is the first of the log-on fields.)

DEVDMP

The DEVDMP processor produces a device copy of the RAD or pack. The is, individual sectors are copied to tape in the order that hey actually occur. DEVDMP is not recommended as a file saver because "files" do not exist in this context. A device dump, however, is useful whenever preventive maintenance 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 DUMP 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 be 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 DUMP STATS
FRES ALL
GAC
```

3. Schedule seven days of INCREMENTALs.

4. After the INCREMENTALs on the seventh day:

```
FSAVE ALL DUMP STATS
GAC
```

5. Repeat steps 3 and 4.

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:

(Use if granule space is so limited that step 3 above fails.)

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 either of the processors FSAVE or 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.

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. 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. 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 HGP reconstruction 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 or boot the PO tape under the files and perform HGP reconstruction.

6. Restore lost files as they are needed.

LOSS OF THE :SYS ACCOUNT

Significant damage to or loss of the :SYS account should be handled by the following procedure:

1. Use the ZAP key-in to remove users from the system.

2. Boot the PO tape specifying S, T, C, P, I.

3. If there is any chance that the file system may have been damaged, request an HGP reconstruction.

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. Boot under files from the PO tape specifying T, C, P, I and request HGP reconstruction.

2. When reconstruction is finished, perform an instant SQUIRREL. This will save all files that can possibly be saved.

3. Restore the entire file system.

The last step is recommended, although it may not be necessary. Depending on errors reported by HGP reconstruction, the system manager should choose whether or not to proceed with normal processing.

(Section 9 has been deleted.)

(Section 9 has been deleted.)

10. SYSTEM GENERATION

INTRODUCTION

The system generation process for the CP-V operating system is performed by service processors. These processors operate as ordinary batch or on-line 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 required in the system generation process.
PASS2	Generates the required dynamic tables for the target 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 to bring up the target system.

Bootstrap operations, including patching operations (PASS0), are described in CP-V/SP Reference Manual, 90 31 13.

Although the processors described in this chapter are collectively termed SYSGEN, they are in fact separate processors. By organizing the process properly and by 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 39 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 XDELTA load module.

```
!JOB
!PCL
COPYALL LT#POX TO DC
    (Load old monitor tape into files)
END
!LOAD  (LMN,XDELTA),(BIAS,EA00),(SL,F),(NOTCB),;
!      (PERM),(ABS),(NOSYSLIB),;
!      (MAP),(EF,;
!      (XDLT),;
!      (SYMTAB),;
!      (MONSTK))
!ASSIGN  M:PO,(DEVICE,9T),(OUTSN,C00)
     (New monitor tape)
!DEF  CP,C00
!FIN
```

Figure 39. Creation of a New System Tape

Commands for PASS2 fall into the following 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, model numbers, 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 of and initial values for line control tables.

   :MON specifies table sizes, CPU type, and buffer pools.

   :SPROCS lists the processors that are shared.

   :GHOST specifies that certain ghost jobs are to be automatically restarted at boot or recovery time.

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.

7. :HANDLERS2 specifies that certain handlers should not be placed in the monitor root but rather should be placed in the unmapped overlay UMOV. PASS2 decides which handlers will not be placed in the root.

8. Multiprocessing

   :SCPU specifies the number and characteristics of slave CPUs.

LOCCT and PASS3 provide a convenient means of retaining 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. These load modules have presumably been created by use of PCL, PASS2, PASS3, or other equivalent processors, 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 target system for a specific installation, the user must know the desired characteristics of that system so the appropriate information may be supplied via the 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? What are their model numbers? 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 are their special handler entry points? What length is required for their command lists?

5. What resources (e.g., 9-track tapes, private disk packs) are available to 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 individual users need to be authorized to use certain special features or will all users be able to use the features?

23. On what machine or machines will the system be booted (e.g., Sigma 6, Sigma 9, Xerox 560)?

24. Should all handler modules reside in the root?

25. Will the target system have transaction processing facilities?

26. Will the target system use remote processing?

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 system having a hardware configuration compatible with CP-V 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 release tape. However, tree tables for nonstandard processors 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 using 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 modules 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 them.

The PASS3 processor reads control commands specifying which LOCCT tables are to be used to define the load structure of CP-V standard modules or user-defined programs.

The user may specify that a given LOCCT table and associated object modules are to be deleted from disk storage after the component object modules have been loaded.

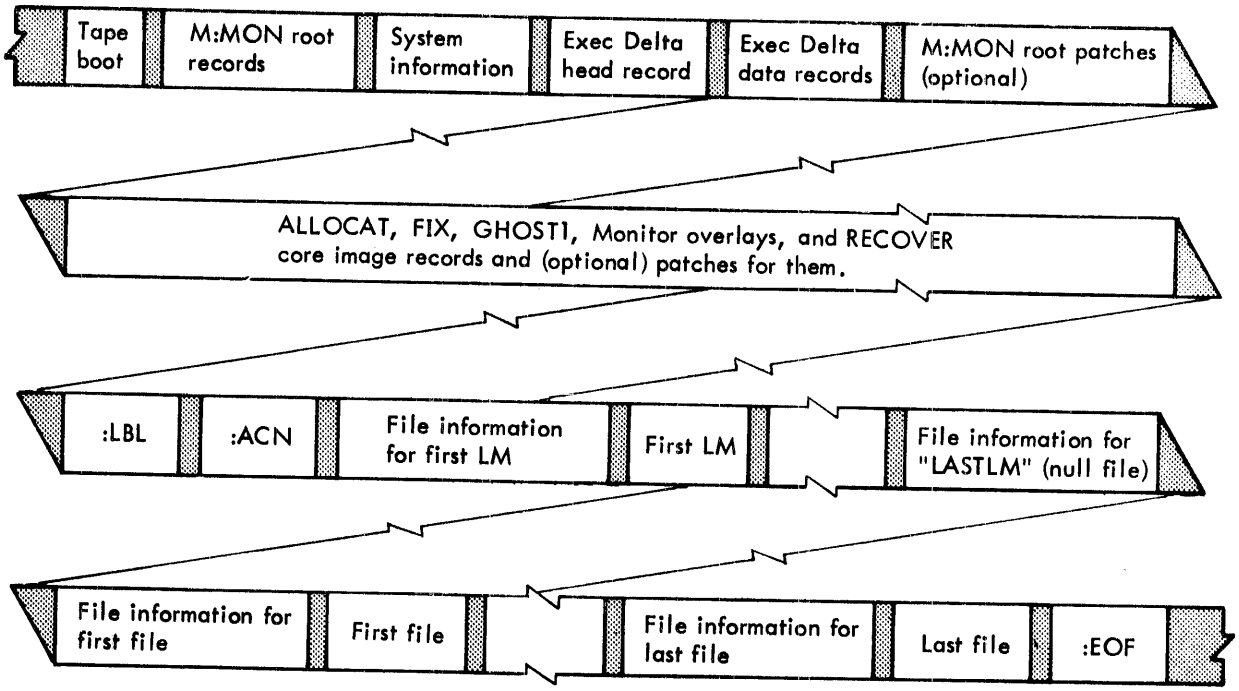
The first command read by PASS3 should specify the monitor's LOCCT table (e.g., M:MON), so that the monitor will be loaded first. Any items loaded will be biased to that bias contained in the LOCCT table for the item (this bias comes from the original LOAD control command used to generate the LOCCT table 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) and RECOVER. M:MON and RECOVER must be loaded by 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 40. 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 a typical CP-V system, and detailed descriptions of the various control commands used in system generation are presented later in this chapter.



Note:

Record sizes — Patch records are 80 bytes long. All other records up to :LBL are 512 words long. The figure indicates groups of such physical records.

Core image records — These records contain data (protection type 00), DCBs (protection type 10), and procedure (protection type 01) for the indicated items except that ALLOCAT has no DCBs and the overlays and RECOVER records have only data. The data records for each load module are preceded by a HEAD record that contains the patching symbol table (see DEF). Optional patches for each protection type follow it on the tape.

Patches — Patches are included on the tape 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 RECOVER patches.

•

Figure 40. Format of Master System Tape

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 that are provided, 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 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 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 (which is typically 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 intermediate 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	:OLIMIT	:SPROCS
:DEVICE	:GLIMIT	:PART
:SDEVICE	:ELIMIT	:FRGD
:LDEV	:IMC	:INTLB
:OPLBLT	:COC	:FAUTH
:RES	:MON	:HANDLERS2
:BLIMIT	:GHOST	:SCPU

With the exception of the :CHAN, :DEVICE, :SDEVICE, :GHOST, :MON, and :COC commands, PASS2 will build dummy commands for those omitted by the user, thereby generating the various load modules with SYSGEN default values. At the end of PASS2, the names of the omitted commands are listed for purposes of information.

Except for :CHAN, :DEVICE, and comment control commands, all other commands should be specified only once. Information requiring more than one card may be included on continuation cards.

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.

All PASS2 name options must consist of alphanumeric characters.

!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

: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_10,n_2c_20))]$_L$,(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 or controllers within this channel 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 B) with NO, MT, and SP being invalid. For a nonstandard device type, the two position yy field is added to the standard device type table 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.)

dddd is the device model number; if unknown, FFFF should be used.

cccc is the controller model number; if unknown, FFFF should be used or field should be omitted.

D specifies that the device is either a RAD or a disk pack. In this case, dddd must be one of the following:

7212, 7232, 3214 — Specify the type of RAD storage device (DC).

7242, 7261, 7271, 7275, 3275 — Specify a disk pack (DP).

The above device numbers are the standard devices recognized by PASS2 for RADs and disk packs.

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 43) 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 43). The names in Table 43 must be used unless a user-supplied handler is to be provided. (See RHANDLER option below.)

When defining a device that is listed in Table 43, 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 43, the HANDLER or RHANDLER 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.

RHANDLER,name1,name2 specifies the name of a user-supplied handler that is to be placed in the monitor root. Name1 is the primary entry and

the name of the module. Name2 is the secondary entry. This option should be used only when all three of the following apply:

1. A :HANDLERS2 control command is included in the PASS2 command set.

2. The handler is a user-supplied handler.

3. The handler is to reside in the monitor root.

In all other cases, the HANDLER option and its provisions apply.

Example:

```
:DEVICE LPA02, (MOD, 7440, 7440)

:DEVICE LPB02, (HANDLER, PRTOUTL, PRTCU), ;
     (MOD, 7450, 7450)

:DEVICE LPC02, (HANDLER, PRTIO2, PRTCU2), ;
     (MOD, 7440, 7440)
```

For device LPA02, the default handler entry names will be used (see Table 43). Otherwise, the handler addresses will be as stated.

When all :DEVICE commands have been processed, PASS2 builds the SPEC:HAND file containing the names of the handlers needed for this particular target machine. This file is used by SYSGEN PASS3. The handlers' 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.

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, MCs, CPs, and RBs. The defaults are: size is 1 except for LPs where size is 26_{16}; widths are X'FF' except for LPs, RBs and MCs where width is 84_{16} and RBs where width is 80_{16} and CPs where width is 78_{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. All of the options listed below must be separately enclosed in parentheses. For example:

```
:DEVICE RBA0A,(MOD,7601),(FULL),(RBX),
  (IRBT),(2780)
```

Table 43. Default Handler Entry Points

Device Type	Model	Name1	Name2
TY		KBTIO	KBTCU
CR	7121,7122, 7140	CRDIN	CRDINCU
CP	7160	CRDOUT	CRDOCU
CP	7165†	CRDOUTL	CRDOLCU
LP	744X	PRTOUT	PRTCU
LP	7446†	NSLP	NSLPCU
LP	7450†	PRTOUTL	PRTCU
LP	3463-3466†	NSLP	NSLPCU
LP	7463-7466†	NSLP	NSLPCU
DC		DISCIO	DISCCU
9T	931X†	MPCTIO	MPCTCU
9T	73XX	MTAP	MTAPCU
9T	3345-3347†	NSTAP	NSTAPCU
DP	9210	MPCDIO	MPCDCU
DP	7271/7242	DPAK	DPAKCU
DP††	7261/7275 3275/3282	DISKAB	DSKABCU
ME	7611	COC	COC
7T	7362,7372	7TAP	7TAPCU
RB		_††	_††
MO	IBM 3270	3270IO	3270CU
MC		RAS	RASCU
PL	7530,7531	PLOT	PLOTCU

†Handlers must be specified.

††The default handlers should be used. The handler entry points are dependent upon the type(s) of remote terminals (see below) or disk device specified.

{7670 / 2780 / 3780 / IRBT} specifies whether the DSC is usable for Xerox 7670 RBTs, IBM 2780 RBTs, IBM 3780 RBTs, or HASP Multileaving IRBTs. IRBT, 2780, and 3780 may be specified for the same DSC in any combination. However, no combination with 7670 is legal. If IRBT and one of 2780 or 3780 are specified, either separately for different DSCs or together on the same DSC, any

type of terminal (IRBT, 2780, or 3780) can be connected to any DSC for which 2780, 3780, or IRBT is specified. In this case, for example, an IRBT may be connected to a DSC which was defined for use with 2780s. Also, a 3780 might be connected to a DSC which had only 2780 specified.

{FULL / HALF} 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.

{RBS / RBX} 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.

DIAL specifies that this is a dial-up line. If DIAL is not specified, any disconnect of the line (e.g., ERROR MAX) will require an RBS key-in to restart the line.

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 and therefore options applying to the set must be placed on the first :DEVICE command of the set. These options should not be used with the standard device types (Table 43).

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. If the HANDLER option is used and NSTAP, NSTAPCU is specified, then the CLIST option is required with a value of 8.

DD specifies dual density feature. It applies only to magnetic tape and indicates that the tape may be read or written at either 800 bpi or 1600 bpi density.

CC specifies code conversion. It applies only to magnetic tape and indicates that the hardware can handle conversion between EBCDIC code in core and ASCII code on tape.

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. This value should be an even number. Otherwise it will be converted by PASS2 to the next lower even number. The maximum value allowed is X'FE'. For nonstandard devices, the default is 6. The default values for standard devices are listed in Table 43-1.

Table 43-1. Default CLIST Values

Device	CLIST Default (Hexadecimal)
TY	6
PR	10
PP	10
CR	2
CP	4A
LP	6
DC	6
9T†	8
7T	8
DP†	8
PL	6
RB	A
ME	6
MO	12
MC	6

†MPC devices have a default hex value of 12.

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. 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. 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.

RAD

When the system is initialized at 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 PSA option may not be used for an MPC disk pack. A control command for a PSA disk pack must precede all other disk pack control commands.

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)

and must not exceed X'FF'. If it exceeds X'FF' PASS2 will reduce it to this limit.

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. PSA may not be allocated on a 9210/9211.

CYLIN, value specifies, in hexadecimal, 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 1 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; that is, it is set to the NCYL value. 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 files on this disk 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 Volume Recognition (AVR) table, and PFA is set equal to SIZE ignoring the options PER, PSA, and PFA.

Table 44 identifies the fixed characteristics and the PASS2 default values of disk devices. All values are expressed in hexadecimal.

Table 44. Standard Device Type Option Default Values (In Hexadecimal)

DISK DEVICE MODEL NO.	SIZE	SS	NSPT	NCYL	NTPC	CYLS	TRKS	SECS	CYLIN
7232	200 tracks	100	C	–	200	–	14	10	–
7212	40 tracks	100	52	–	40	–	17	10	–
3214	100 tracks	100	B	–	100	–	14	10	–
7242	C8 cyls.	100	6	C8	14	10	8	–	1E
7261	C8 cyls.	100	B	C8	14	10	8	–	37
7271	190 cyls.	100	6	190	14	10	8	–	1E
7275	194 cyls.	100	B	194	13	10	8	–	34
3275	194 cyls.	100	C	194	13	10	8	–	39
9210†	336 cyls.	100	8	336	1C	10	8	–	7A

†A 9210 model number implies an MPC disk pack.

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. Standard disk devices are recognized by their model numbers, as listed in Table 44. Note that the previously defined options SS, NSPT, and SIZE should also be used for defining nonstandard disk devices unless the default values apply.

{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. The default is 10_{16}.

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. The default is 8.

SECS, value specifies, in hexadecimal, the shift factor for the sector portion of the disk addresses. This parameter applies only to fixed head disk devices. The default is 10_{16}.

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 nor 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, SP is equivalent to the first disk pack device which was mentioned (MOVE specified on :DEVICE card).

: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[, KEEP][, OPCON] 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. KEEP specifies that all granules of the symbiont file for this device

are to be kept until output has been completed. OPCON specifies that the operator is to have complete control over this symbiont device. Automatic restart will not occur. Restart will take place only after an appropriate operator key-in.

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.

Example:

```
:SDEVICE (OUT, LPA02), (OUT, LPA0F, KEEP), ;
    (OUT, CPA04, KEEP, OPCON), ;
    (OUT, XPA06, OPCON, KEEP), (IN, CRA03)
```

: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.

Example:

```
:LDEV (C2,CR),(L2,LP),(L3,LP),(P2,CP)
```

: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, MC, 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 same restrictions as 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 same restrictions as 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 45. (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.

Example:

```
:OPLBLT (GI,MEA10,ME,TYA01)
```

Table 45. 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

Table 45. Standard Default Operational Labels and Assignments (cont.)

Standard Operational Label	Batch	On-Line	Ghost
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 or 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. All options preceding the first RES option are ignored. Four resources are established by default whether or not a RES option is specified--CO, 9T, 7T, and SP. However, 9T, 7T, and SP will only be established by default if at least one such device appears on a :DEVICE command. These resources may be specifically established via the :RES command if the default values are not desired. (This is described shortly.) If an MC device was defined on a :DEVICE command, an MC resource is also automatically established. Therefore, MC should not be specified on the :RES command. In addition to these default resources, other resources may be defined. A system may have a total of 15 resources.

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. Specifying OSUM for COre is meaningless.

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. One GDEF for tape is required at boot time. If the user specifies a zero value, PASS2 will change it to 1.

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 first four resources listed in Table 46 with the default values as shown except when they are specified on the :RES command with other values specified. PASS2 generates the fifth resource (MC) if an MC device was specified on a :DEVICE command. MC cannot be specified on the :RES command.

Table 46. :RES Command Defaults

Option Name \ Resource Name	CO (core)	9T (tapes)	7T (tapes)	SP (private disk packs)	MC① (remote assist terminal)
TOT	X'7FFF'	#②	#	#	1
BSUM	X'7FFF'	#	#	#	1
BMAX	X'10'	#	#	#	1
BDEF	X'C'	0	0	0	0
OSUM	X'7FFF'	#	#	#	1
OMAX	X'10'	#-1③	#-1③	#-1③	1
ODEF	X'C'	0	0	0	0
GSUM	X'7FFF'	#	#	#	1
GMAX	X'FF'	#-1③	#-1③	#-1③	1
GDEF	X'FF'	#-1③	#-1③	#-1③	1

① MC is included only if an MC device has been specified.

② # is the total number of this type of device specified on :DEVICE commands.

③ 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 / :GLIMIT} [(option)][,(option)]...

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.

Table 46-1. :BLIMIT, :OLIMIT, :GLIMIT Command Default Values

	BLIMIT		OLIMIT		GLIMIT	
	DEFAULT	MAXIMUM	DEFAULT	MAXIMUM	DEFAULT	MAXIMUM
TIME	5	$2^{31}-1$	$2^{31}-1$	$2^{31}-1$	$2^{31}-1$	$2^{31}-1$
LO	100	$2^{15}-1$	100	$2^{15}-1$	$2^{15}-1$	$2^{15}-1$
PO	100	$2^{15}-1$	100	$2^{15}-1$	$2^{15}-1$	$2^{15}-1$
DO	500	$2^{15}-1$	100	$2^{15}-1$	$2^{15}-1$	$2^{15}-1$
UO	100	$2^{15}-1$	100	$2^{15}-1$	$2^{15}-1$	$2^{15}-1$
PSTORE	64	$2^{31}-1$	64	$2^{31}-1$	$2^{31}-1$	$2^{31}-1$
TSTORE	64	$2^{31}-1$	64	$2^{31}-1$	$2^{31}-1$	$2^{31}-1$
FPOOL	6	127	4	127	64	6
TDISK	64	$2^{31}-1$	64	$2^{31}-1$	$2^{31}-1$	$2^{31}-1$
PDISK	64	$2^{31}-1$	64	$2^{31}-1$	$2^{31}-1$	$2^{31}-1$

: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 except for BPRIO, OPRIO, GPRIO, and SECURITY):

MAXOL, value specifies the maximum number of on-line users allowed in the system. MAXOL must be in the range $255 \geq MAXOL \geq 0$. The default is 0. (OUM)

MAXB, value specifies the maximum number of concurrent batch users allowed in the system. MAXB must be in the range $16 \geq MAXB \geq 1$ but cannot exceed the number of batch partitions defined by the :PART command. The default is 1. (BUM)

MAXG, value specifies the maximum number of ghost jobs (excluding GHOST1) allowed in the system. MAXG must be in the range $255 \geq MAXG \geq 10$. The default is 10. (GUM)

The first five ghost jobs are reserved for system required jobs, therefore the value specified should include an allow-

ance for this minimum number plus the user's desired number of ghost jobs.

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. (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. (OPRIO)

GPRIO, value specifies the execution priority for all ghost jobs. This priority is used for execution only and is not related to the scheduled priority. The value must be expressed in hexadecimal and must be in the range C0-FF. The default is FC. (GPRIO)

BLOCK, value specifies the number of seconds of output at which terminal is unblocked. The default is 1. The maximum value is 32,767. (TB)

UNBLOCK, value specifies the number of seconds of output at which terminal is unblocked, where UNBLOCK ≤ BLOCK. The default is 1. The maximum value is 32,767. (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. 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. The maximum value is 655. (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. The maximum value is 655. (OITO)

PI, value specifies the increment in units of 1/256 to be used in increasing the priority of jobs by-passed by the Multi-Batch Scheduler. The default is 0. (PI)

SECURITY, value specifies a hexadecimal constant (up to eight digits long) to be used in initializing core pages allocated to a user for whom the Super SECURITY option has been specified. A value of less than eight digits will be right-justified. The default is 0.

UCYL, value specifies the number of physical 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.

AIRM, value specifies the maximum number of AIR (File Directory) granules to be held in core at one time, where AIRM ≤ RAMAX. The recommended value is 4. The default is 0. (AIRM)

AIRTO, value specifies the time (in milliseconds) after which an unused AIR block will be purged (timed-out). The value must be in the range of 0 to 500000. The default is 30000. (AIRIO)

RASIZE, value specifies the number of table entries to be built for use by read-ahead and AIR. One table entry is required for each concurrent read-ahead operation and one for each active AIR granule. 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 value must be in the range 0-63. 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)

PWP, value specifies the maximum number of core pages that can be used as physical work pages in a transaction processing environment. A minimum of 12 is suggested. A default valve is not automatically provided. Six of these pages will be reserved for system use, thus reducing the maximum user size by that amount.

RETIM,value specifies, in minutes, the amount of time an on-line user's swap image is to be retained after the user has been disconnected due to a line hang up condition. The value must be in the range 0 to 65,535. The default is 0. (RETIM)

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 limit.

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 absence of a :COC command causes a batch-only system to be generated (i.e., there will be no COC handler). 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 where $0 \leq n \leq 8$).

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.

CALL360 specifies that the CALL360 translation table is to be included in the monitor for a 2741 terminal.

ASCIIAPL specifies that a translation table for ASCII terminals with APL character sets is to be included in the monitor. This table is for Tektronix graphics scopes, Diablo-based printers (such as the Xerox 3010), and similar terminals.

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.

One of the following two device options must be specified:

L6 specifies that this device is a Level 6 FECP (Front End Communications Processor). Up to 2 FECPs are allowed in the system. Specification of any FECPs on the :COC command must follow the specification of any normal COC devices.

DEVICE,ndd specifies, in hexadecimal, the device address of a COC device.

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 and output interrupts for all COC devices must be within one and the same interrupt group.†

Each COC input interrupt should be one higher in priority than the corresponding COC output interrupt. 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).

LINES,value specifies, in decimal, the number of lines. The default is 8. The maximum for each COC specified is 128.

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.

®Registered trademark of the International Business Machine Corporation.

†External interrupts are described in the Sigma and Xerox 560 hardware reference manuals.

ECB,value specifies, in decimal, the number of buffers to be used for Event Control Block Processing and Transaction Processing. Requirements for TP are 1 for each MO device, 1 for each COC slave line, and 10 for ECBs. System requirement is a maximum of 1 per QUEUE value (:MON control command) for I/O associated with an ECB. A default value is not automatically provided.

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.

AUTO specifies that the COC hardware is to be interrogated and the line configuration determined automatically. The information normally supplied by the 2741, TYPE, and RATE keywords will be set up at COC initialization time, except for lines that have a user associated. If 150 baud lines exist, AUTO should not be used, as they will be set up as 2741 lines.

D7015,#,... 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.

D33,#,... specifies, in decimal, the line numbers that are attached to a Model 33 Teletype. Each # field may be a single value or a range specification of m-n, where m < n. For a range, all numbers included in the range are used as line numbers. Any line numbers associated with this option must also be declared as 2741-line numbers within the 2741 keyword option.

D35,#,... specifies, in decimal, the line numbers that are attached to a Model 35 Teletype. Each # field may be a single value or a range specification of m-n, where m < n. For a range, all numbers included in the range are used as line numbers. Any line numbers associated with this option must also be declared as 2741-line numbers within the 2741 keyword option.

D37,#,... specifies, in decimal, the line numbers that are attached to a Model 37 Teletype. Each # field may be a single value or a range specification of m-n, where m < n. For a range, all numbers included in the range are used as line numbers. Any line numbers associated with this option must also be declared as 2741-line numbers within the 2741 keyword option.

DASCIIAPL,#,... specifies, in decimal, the line numbers that are attached to an ASCII terminal with an APL character set. Each # field may be a single value or a range specification of m-n, where m < n. For a range, all numbers included in the range are used as line numbers. Any line numbers associated with this option must also be declared as 2741-line numbers within the 2741 keyword option.

DEA,#,... specifies, in decimal, the line numbers that are attached to a Model 2741 with EBCD APL typeball. Each # field may be a single value or a range specification of m-n, where m< n. For a range, all numbers included in the range are used as line numbers. Any line numbers associated with this option must not be declared as 2741-line numbers within the 2741 keyword option.

DES,#,... specifies, in decimal, the line numbers that are attached to a Model 2741 with EBCD standard typeball. Each # field may be a single value or a range specification of m-n, where m < n. For a range, all numbers included in the range are used as line numbers. Any line numbers associated with this option must not be declared as 2741-line numbers within the 2741 keyword option.

DSA,#,... specifies, in decimal, the line numbers that are attached to a Model 2741 with Selectric APL typeball. Each # field may be a single value or a range specification of m-n, where m < n. For a range, all numbers included in the range are used as line numbers. Any line numbers associated with this option must not be declared as 2741-line numbers within the 2741 keyword option.

DSS,#,... specifies, in decimal, the line numbers that are attached to a Model 2741 with Selectric standard typeball. Each # field may be a single value or a range specification of m-n, where m < n. For a range, all numbers included in the range are used as line numbers. Any line numbers associated with this option must not be declared as 2741-line numbers within the 2741 keyword option.

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. Any line numbers associated with this option must not be declared as D7015 lines.

HARDWIRE,#,... specifies, in decimal, the line numbers that are hardwired to terminals. Lines which are "hardwired" will not be timed out (except 2741s logging on), nor will they log on automatically. All lines must be specified but need not be in ascending order. Each #field may be a single value or a 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 5. The seven timing algorithms that are supplied with CP-V are listed in Table 47. The default TYPE value for lines that are listed on the 2741 option is 1. The default TYPE value for all other lines is 5. All line numbers may be specified, but need not be in ascending order. Each # field may be a single value or a 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 of 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 specified, but need not be in ascending order. Each # field may be a single value or a specification m-n where m<n. In the latter case, all numbers included in the range are used as line numbers.

HD,#,#... specifies, in decimal, the line numbers that are attached to half-duplex modems. 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. Half-duplex lines should be installed so that the line numbers are as close to being one contiguous group as possible.

COCDIO,value where value overrides the default direct I/O (DIO) address for this COC. Normally, the DIO address is calculated as the number of this COC minus one. Thus, in the case of four COCs, the addresses would normally range from 0 to 3. When a COCDIO value is specified, subsequent COC addresses will be incremented by 1 beginning with the value specified until another such option is encountered. Using the example of four COCs and given that a COCDIO value of 3 was specified for the second COC, the four addresses would be constructed as follows: 0, 3, 4, 5.

COUPLE specifies that the terminal coupling feature is to be included in the system. Specifying COUPLE causes the TIE table to be generated; the TIE table is 1 byte per line in size. This feature is inoperative if MINICOC is selected by the :MON command.

Terminal translation types are determined by the options D7015 through DSS. D7015 through DASCIIAPL are non-2741 terminal type options. DEA through DSS are 2741 terminal type options.

Whenever a line disconnects or connects, or when a hard-wired line logs off, the default terminal type becomes the current type. If a dialup line logs off and back on without disconnecting, the type of terminal that was in effect at log-off time remains in effect for the new session.

The current terminal type can be changed by the user via the TEL TERMINAL command or the M:STA monitor service call. The default terminal type can be specified on a line-by-line basis by using the appropriate terminal translation type. If a line's default type causes a 2741-conflict, a diagnostic message is produced. When a line's default type is not specified and AUTO is not specified, the default is Teletype Model 33 for non-2741 lines and unidentified for 2741 lines. (Unidentified requires the terminal user to type an asterisk (*) when logging on.) If a line's default type is not specified and AUTO is specified, the default is determined by line speed/format as follows:

Speed or Format	Default Type	COCG DEF
110 Baud	Teletype Model 33	Type 110
300 Baud	Teletype Model 37	Type 300
600 Baud	Teletype Model 37	Type 600
1200 Baud	Teletype Model 37	Type 1200
2400 Baud	Teletype Model 37	Type 2400
2741 Format	"unidentified"	Type 2741

: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. The value n must be in the range 1-64.

SIG5 specifies that the target system is a Sigma 5.

SIG6 specifies that the target system is a Sigma 6. The keyword SIG7 may be used in place of SIG6 and has the same meaning.

SIG9 specifies that the target system is a Sigma 9.

X560 specifies that the target system is a Xerox 560.

BIG indicates that the target system can have more than 128K of memory. BIG may be used in conjunction with either a SIG9 or X560 option but not with both.

MOS indicates that the target system has MOS memory. MOS may not be specified in conjunction with X560. MOS may be specified for a Sigma 9 that has both MOS memory and core memory.

(This page intentionally left blank.)

Table 47. 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 none
After carriage return (curpos††+15)/10
After tab character (new position-old position+15)/10</td></tr>
<tr><td>2</td><td>Execuport, Dataport, and TI Model 33 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>21-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 than in 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>21-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>17</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>
<tr><td>6</td><td>Teletype Model 40 hardcopy printer.</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>61-120 cps</th><th>121-240 cps</th></tr>
<tr><td>Before carriage return</td><td>3-curpos††</td><td>5-curpos</td><td>9-curpos</td><td>17-curpos</td><td>34-curpos</td><td>67-curpos</td></tr>
<tr><td>After carriage return</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>After tab character</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</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.

The target system specification causes the appropriate machine-specific module of code to be automatically included in the monitor root by SYSGEN. The following combinations are valid target system specifications:

(SIG5)	(SIG5),(SIG9)	(SIG5),(SIG9),(X560)
(SIG6)	(SIG5),(MOS)	(SIG5),(MOS),(BIG)
(SIG9)	(SIG6),(SIG9)	(SIG6),(SIG9),(X560)
(X560)	(SIG6),(X560)	(SIG6),(SIG9),(MOS)
	(SIG9),(X560)	
	(SIG9),(BIG)	(SIG6),(SIG9),(MOS),(BIG)
	(X560),(BIG)	

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. If the use of concurrent output mode is anticipated (refer to the CONCURR option of the LDEV control command in the CP-V/BP Reference Manual), OUTFILE must be set high enough to accommodate the additional output symbiont file slots required. 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 RBT line and by a number that is approximately three-fourths the maximum number of peripheral devices that could exist at concurrently logged on IRBTs.

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 700.

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. 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.

TP specifies that the transaction processing facility is to be a part of the target system.

MINICOC specifies that the MINCOCR/MINCOCU handler is to be used instead of the TPCOCR/TPCOCU or REGCOCR/REGCOCU handler. If TP has been specified, the MINICOC specification will be ignored and TPCOCR/TPCOCU will be used.

HOTAVR specifies that ARV processing is to be automatically performed on any write-protected tape that is in a ready state and occupies a 'not-in-use' tape drive. The use of this option causes the inclusion of the HOTAVR handler in the UMOV monitor overlay.

HOTCARD specifies that any card reader symbiont devices that are inactive, but operational and ready, are to be automatically started by the system. This option simulates an operator key-in of !SCRnnd,1.

(This page is intentionally blank.)

istics of shared processors or installation-created monitor overlays and optionally reserves space for the dynamic addition or replacement of shared processors. The :SPROCS command has the form

:SPROCS [(name[,option]...)]$_L$,(name[,option]...)]...

where name may be

MON followed by the names of installation-created overlays to be added to the list of standard, default overlays which are KEYIN, LTAPE, DEBUG OPEN, CLOSE, STEPOVR, LDLNK, MULOV, MISOV, RMAOV, UMOV, ECBOV, ENQOV, RTOV, TQOV1, TQOV2, and OPENTP. The names must be less than or equal to seven characters in length and must not be ENQ or RTROOT. Only those portions of the monitor that have been specifically coded as overlays must be specified. This is necessary because of the mapped overlay scheme that is used. The names of these user-created overlays must be added to the standard CP-V monitor LOCCT and code must be added to the TOPROOT or T:OV modules to access these overlays. Use of CP-V standard overlay names is not required by PASS2. Overlays automatically included in the target system as a function of PASS2 control command options are ECBOV, ENQOV, RTOV, UMOV, TQOV1, and TQOV2. The PASS2 options that control this automatic overlay mechanism are listed in Table 49.

processor name followed by flags and/or a number in the form

(procname,flags,number)

or

(procname,flags)

(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 ($0 \leq i \leq 9$). 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.

X specifies that this processor has M:SYS CAL authorization.

Table 49. PASS2 Options That Control Automatic Overlays

System Feature	PASS2 Option	Overlay(s) Included
Transaction processing†	TP on :MON command	TQOV1, TQOV2
Real-time	:FRGD command	RTOV
ENQ/DEQ†	ENQ on :MON command	ENQOV
Unmapped monitor overlay requirement	:HANDLERS2 command	UMOV
Target system is an MPC disk pack or tape	(MOD,D,[9210 / 9310]) on :DEVICE command	MPC9210, MPC9310

†If neither transaction processing nor ENQ/DEQ is requested, then ECBOV is automatically included.

The flags S, J, D, P, M. T, B, G, C, and X may also be grouped syntactically without intervening commas (e.g., SJ, etc., up to a maximum of eight characters). If D or P is specified, S is redundant.

Note: The first eight processors will be monitored for STATS.

Table 50 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)

Table 50. Standard List of Shared Processors

Processor Name	J	S	D	P	M	T	G	B	X	No. of Overlays
TEL	1	1	0	0	1	1	0	0	1	0
CCI	1	0	0	0	1	0	0	1	0	0
FIX	0	0	0	0	1	0	0	0	0	0
GHOST1	1	0	0	0	1	0	1	0	0	0
ALLOCAT	0	0	0	0	0	0	0	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
DRSP	0	0	0	0	1	0	0	0	0	0
RUNNER	1	1	0	0	0	0	0	0	0	0

Procname, flags, and number are described above. The processors in Table 50 can be specified on the :SPROCS command only in conjunction with REPLACE.

For those installations in which processors are to be dynamically added or replaced, the following name and options will be required as applicable:

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., 110 for the 7260/7275 or 110 for the 7261/7275). 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)
```

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, four slots have been reserved for shared processors each of which will have 64 pages reserved for them and four slots have been reserved in the processor overlay area for additions or replacements.

: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 resouce 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). A resource name is always generated for C0 (core) and for the four remaining standard resources – 9T, 7T, SP, and MC – providing they have been previously defined. The minimum values default to zero. The maximum values default to 1. A total of fifteen resouce names is permitted including the number of standard resouces defined.

{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.

The following three options specify the origin of jobs which may be run in the partition. If none of these options is specified, all three will be assumed by default.

LCL specifies local batch.

TRM specifies on-line terminal.

RB specifies remote processing.

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, address

where

size specifies, in decimal, size in pages of the dedicated foreground memory area to be allocated at system initialization. The size may range from 0 to 999. The default is 10.

address specifies the word address of the first page in the RESDF memory segment. This value must be equal to or greater than 10000_{16}. The default is 10000_{16}.

Both size and address may be overridden by the operator at system initialization. Both parameters may be reset via communication with the Physical Page Stealer (PPS) ghost job.

DYNRESDF, pages, segments

where

pages specifies, in decimal, the maximum number of pages that may be dynamically allocated for foreground use. These pages are not removed from the system until requested, but the maximum user size is reduced by the value specified. This value may be altered by the operator via the PPS ghost. The value specified may range from 0 to 999. The default value is 0.

segments specifies, in decimal, the maximum number of dedicated real-time memory segments that may be allocated for foreground use. The value may range from 1 to 999. The default value is 1.

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, prio)[, (label, loc, prio)] . . .

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 60-13F. The permissible values for a pseudo-interrupt number are in the hexadecimal range 1000-7FFF.

prio specifies, in hexadecimal, the default execution priority to be associated with this interrupt label.

: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

TP - transaction processing

AS - auto-save on line disconnect

CJ - common journalizing

WC - concurrent outputting

:HANDLERS2 The :HANDLERS2 command specifies that the destination of certain handlers is to be the UMOV overlay instead of the monitor root. (UMOV is a resident "overlay" in which all code executes unmapped.) The generation of large target systems may require the utilization of this command to remove code from the monitor root. :HANDLERS2 must be specified if an MPC device is specified on a device control command. Table 51 indicates which handlers are candidates to be placed in UMOV when the :HANDLERS2 control command is specified. Also, when the :HANDLERS2 command is specified, PASS2 builds a ROOTHAND module. If :HANDLERS2 command is absent from the control command input, all required handlers are placed in the monitor root. The format of the :HANDLERS2 command is:

:HANDLERS2

Table 51. Distribution of Handlers When :HANDLERS2 Control Command is Specified

Monitor Root	UMOV
EAPL	BSCIO
ESTD	CRDIN
HSPM	CRDOUT
MOCIOP	CSEHAND
MPSUB	CSECOM
PWP	CSES7
RA	CSES9
RAS	CSEX560
RTROOT	DISCIO
SAPL	DISKAB
SCHDSUB	DPAK
SFAULT	DSCIO
S5SIM	DPSIO
SMON	HASPIO
SSTD	HOTAVR
TPCOCR/REGCOCR/MINCOCR†	HOTCARD
TQROOT	INITRCVR
User-supplied handlers††	INSYM
AAPL	IOQ
CALL360	KBTIO
	MAGTAPE
	MPCDIO
	MPCTIO
	MPSCHED
	NSLP
	OCPIO
	OUTSYM
	PLOT
	PRTOUT
	PTAP
	RBSSS
	SSDATU
	SWAPPER
	TPCOCU/REGCOCU/MINCOCU†
	TSIO
	V2IO
	2780IO
	3270IO
	7TAP
	User-supplied handlers††

†Only one of these handlers is used.

††These handlers are dependent on specification of the RHANDLER or HANDLER option on the :DEVICE command.

: SCPU This control command establishes the existence of a multiprocessing system and defines the number and characteristics of the secondary central processing units. The labels in parentheses are those associated with the corresponding variables by the Control processor. The format of the :SCPU command is

:SCPU (NSCPU, n), (INTS, value[, value[, value]])
[, (option)]...

where

NSCPU,n specifies the number of secondary CPUs on the system. The value n may range from 1 to 3. (If n is 1, then secondary CPU number 1 has been defined; if n is 2, secondary CPUs numbered 1 and 2 have been defined, etc.)

INTS, value[, value[, value]] specifies the values in hexadecimal, of the secondary to primary CPU interrupt levels. One value must be specified for each of the secondary CPUs defined by NSCPU. The first value applies to secondary CPU number 1, etc.

The options are:

MPIPI indicates that the secondary to primary interrupt level references an interrupt pair. If not specified, then it is a single interrupt configuration.

AUTO, n[, n[, n]] specifies the numbers of the secondary CPUs which are to be automatically started at boot or reboot time.

NOAUTO specifies that automatic starts at boot and reboot time are to be overridden. It is important to note that NOAUTO may be changed dynamically using Control. (BSTR)

MINQ, value[, value[, value]] specifies, in decimal number of milliseconds, a guaranteed time slice for a user on a secondary CPU after a selection has been made and before a swap will occur. The first value is applied to secondary CPU number 1, etc. If only one value is specified, the value is applied to all secondary CPUs defined by NSCPU. The range of permissible values is 10 to MAXQ for the corresponding secondary CPU. The default is 100. (MINQ1, MINQ2, MINQ3)

MAXQ, value[, value[, value]] specifies in decimal number of milliseconds, the maximum time slice to be allocated by the secondary CPU to a user. The first value is applied to secondary CPU number 1, etc. If only one value is specified, the value is applied to all secondary CPUs defined by NSCPU. The range of permissible values is MINQ for the corresponding secondary CPU to 32767. The default is 2000.

:GHOST This control command is used to define the names of ghost jobs to be automatically started when booting from the system device or following a recovery. The system currently provides this service for the ALLOCAT, RBBAT, FIX, KEYIN, FILL, ERR:FIL, RVGHOST, and ANLZ ghost jobs. Additions to this list may be made by using this command, which has the following format:

:GHOST(name[,[account][,priority]])[,(name...)]...

where name must be seven characters or less, account must be eight characters or less, and priority must be within the range of C0 to FE. The default account is :SYS and the default priority is GPRIO (specified in the :IMC command).

For example:

```
:GHOST(LOGON),(LINK,CPVLMA),(RUNNER,
        ACCTCPV,FD),;
:       (LDEV,,FE)
```

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 42 is the functional tree structure of PASS2 (the names identify the necessary element files).

LOAD MODULES AND FILES GENERATED BY PASS2

Table 52 identifies the load modules, data files, or element files† that are generated by the corresponding PASS2 control commands.

PASS2 MESSAGES

All PASS2 messages (Table 53) 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).

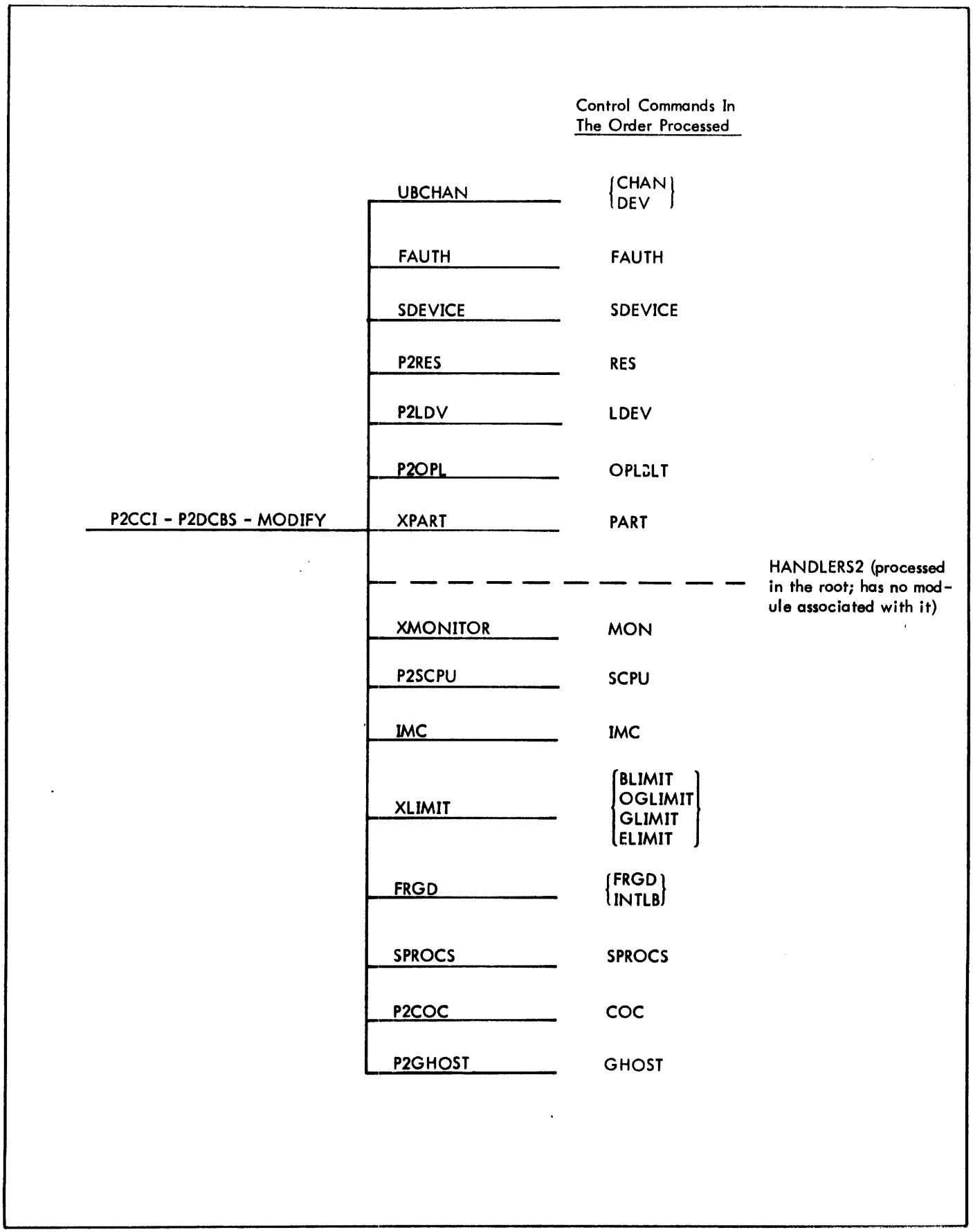


Figure 42. Tree Structure of PASS2

Table 52. 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)
FAUTH	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:CONFIG (load module) MON::ORG (element file) CSEBRANCH (load module) SPEC:HAND† (data file)
SCPU	STABLES (load module)
{BLIMIT GLIMIT OLIMIT}	SG:LNT (load module) SG:DLNT (load module)

Table 52. PASS2 Load Modules and Files (cont.)

PASS2 Commands	Load Module/Element File Name
ELIMIT	M:ELIMIT (load module)
COC	M:COC (load module) M:IOMOD† (load module) SPEC:HAND† (data file) SG:INT† (load module) ROOTHAND(load module)††
IMC	M:IMC (load module)
SPROCS	M:SPROCS (load module)
PART	M:PART (load module)
{FRGD INTLB}	SPEC:HAND† (data file) M:FRGD (load module)

†These modules are built from information from several commands. In addition to the above files, PASS2 also generates the T:P2SI file which contains all the control commands used during PASS2 and a list of any significant errors that are not of the abort type. This file is accessed by the monitor dump analysis program (ANLZ) and is included with its output to aid in system debugging.

††This module is built only if a HANDLERS2 command has been specified.

Table 53. 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.
***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 DUPLICATE IGNORED	A duplicate control command other than a :DEVICE or :CHAN command was encountered. PASS2 ignores duplicates and continues.

Table 53. PASS2 Messages (cont.)

Message	Description
..... 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 IN PROCESSOR JOB ABORTED	One of the control command modules has transmitted incorrect syntax parameters to the syntax routine.
***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.
***ERROR WHILE WRITING T:P2SI	An I/O error occurred. PASS2 continues.
######xx ERRORS AND OR WARNINGS IN THIS PASS2######	PASS2 keeps a count of the error and warning messages it has issued and prints the number in the xx field.
***INADEQUATE CORE SPACE – SKIP TO NEXT CC	The current option caused a table overflow.
***INADEQUATE SPACE TO BUILD INTER-MEDIATE TABLES ***PASS2 ABORTS	An attempt to get a page for use as a work area has failed. PASS2 aborts.
***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.
***MODIFY ERROR – SKIP TO NEXT CC ***THE FOLLOWING TABLE WAS BEING GENERATED. xxxxxxx	Problem generating a DEF or REF (xxxxxxx) during the building of the current load module. PASS2 aborts.
***NO COC COMMAND – A BATCH ONLY SYSTEM BEING GENERATED	Self-explanatory. PASS2 continues.
PASS2 AT YOUR SERVICE	This is an on-line message to inform you that PASS2 is ready to read SI and start processing.
.....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 non-affected 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.
***SYNTAX ERROR – ...	The rest of the message indicates the type of error encountered.
THE FOLLOWING COMMANDS WERE NOT FOUND IN THE INPUT TO THIS PASS2: :xxxxxxx	This message lists any of the 20 possible commands not included in the input to PASS2. This is an informative message only.

Table 53. PASS2 Messages (cont.)

Message	Description
***UNKNOWN OR MISPLACED CC	The current control command is unknown or out of order. PASS2 continues to next control command.
CHAN/DEVICE Messages	
$	When $ appears without additional messages, it indicates that there is a syntax error. PASS2 tries to continue.
***A PSA-ALLOCATED PACK MUST ALWAYS PRECEDE ALL OTHER PACK DEVICE COMMANDS (DELAYED ABORT)	The PSA pack device control command must physically precede all other disk pack commands.
***A STANDARD DEVICE CLIST VALUE CANNOT BE LESS THAN SYSGEN DEFAULT. SYSGEN DEFAULT USED	User has specified a CLIST value less than the minimum allowable for the device.
***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 "NO", "MT", or "SP" as its yy. PASS2 tries to continue.
***DP INDEX EXCEEDS 63 DEC. THIS IS A PROBLEM IF 3282 PACKS ARE INVOLVED.	The user should rearrange input so that 3282 packs will have a lower DCT index than other packs. PASS2 continues.
***FIXED HEAD DEVICE EXCEEDS MAX TRACKS FOR SWAPPER – SET TO MAX TRACKS	A PSA value greater than 200 tracks has been specified for a non-standard RAD device.
***IF SS, TRKS, CYLS OMITTED DEFAULTS: 100, 8, 10 WILL BE SUPPLIED	This message applies to nonstandard disk or disk pack devices.
***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.
***MORE THAN 36 DP-3275 CHANNELS SPECIFIED ONLY FIRST 36 RETAINED	There are over 36 channel control commands associated with 3275 model disk packs. Only the first 36 will be retained for later use by the XMONITOR module in PASS2. PASS2 continues.
***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.
***NCYL MISSING FOR MOVEABLE HEAD DISK DEVICE	In defining a nonstandard disk device, the parameter NCYL was omitted. PASS2 aborts.
***NGC >255 – NGC SET TO 30	A CYLIN value greater than 255 was specified for a new device. PASS2 continues.
***NGC >255 – 55 USED FOR 7261/7275/3275	The CYLIN value is too large. PASS2 continues.
***NGC >255 – 30 USED FOR 7242/7271	The CYLIN value is too large. PASS2 continues.
***NGC >255 – X'70' USED FOR 9210	The CYLIN value is too large. PASS2 continues.

Table 53. PASS2 Messages (cont.)

Message	Description
***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.
***NON-STANDARD DEVICE OPTIONS SUCH AS VFC, BIN, R, AND COMP IGNORED FOR STANDARD DEVICES	PASS2 uses a set of standard values for any standard device. Therefore these options should not be specified on a standard device control command.
***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 1 MC-DEVICE COMMAND IS LEGAL	Self-explanatory. 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.
***PASS2 UNABLE TO CONTINUE	Self-explanatory. PASS2 aborts.
***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 53. PASS2 Messages (cont.)

Message	Description
***PSA MUST BE 7212/7232/3214 DISC PACK - PSA IGNORED	PSA has been specified on a device that is not recognized as either a RAD or a disk pack. PASS2 continues.
***PSA NOT VALID ON MPC DP DEVICE. PASS2 ABORT ERROR	The PSA option is not permitted for an MPC device.
***PSA PREVIOUSLY DEFINED ON DP - PSA IGNORED	The PSA option may only be specified for one disk pack device. PASS2 continues.
***PSA VALUE TOO LARGE. REDUCED TO HEX FF.	This message refers to a pack swapper. PSA size for a pack swapper must not exceed one byte. PASS2 continues.
***SINCE AN MPC WAS SPECIFIED, A HANDLERS2 CONTROL COMMAND MUST BE INCLUDED IN THE PASS2 (DELAYED ABORT)	A device control command for an MPC disk pack requires that a HANDLERS2 control command be present in the PASS2 source input.
***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 NONZERO	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	PASS2 will increase PFA, if it exists, to utilize all device tracks on cylinders. If PFA does not exist, then an attempt is made to add the balance to PER. If PER does not exist, then the balance is added to PSA.
***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.
***THE UNIT ADDRESS OF AN MPC DEVICE CANNOT BE ZERO (i.e., THE LOW ORDER D IN THE YYNDD DEVICE NAME).	The low order D of an MPC device must be greater than zero.
***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.
***THIS HANDLER – xxxxxx – WILL BE PLACED IN THE UMOV OVERLAY	The xxxxxx handler has been declared a monitor root handler via the RHANDLER option. However, it is one of the standard handlers destined for the HANDLERS2 record in SPEC :HAND. Therefore, the declaration will be ignored.

Table 53. PASS2 Messages (cont.)

Message	Description
***TOO MANY COC'S SPECIFIED – ONLY 8 ARE ALLOWED	Self-explanatory. PASS2 continues.
***TOO MANY LIKE SWAP DEVICES TO FIT IN S#XXXX TABLE. ONLY FIRST 8 SUPPLIED.	The table of swap device model numbers is only eight words long. PASS2 continues.
***TROUBLE WITH M:MODNUM FILE OR MODNUM COMMAND. PASS2 UNABLE TO CONTINUE.	An I/O error occurred. PASS2 aborts.
***UNKNOWN DEVICE yyndd FOR CH	The channel designated is out of range.
***UNRECOGNIZABLE UNIT ON DEVICE CONTROL COMMAND	An unusual unit number was specified for a non-FECP device. PASS2 changes it to X'1E' and continues.
***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. #. FFFF 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 FOR 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 ***PASS2 UNABLE TO CONTINUE	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.

Table 53. PASS2 Messages (cont.)

Message	Description
***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 46. 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.
***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 46. 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 RESOURCE UNDEFINED AMONG DEVICES	A DP resource was specified but no disk packs were defined via :DEVICE command.
***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 46. For all other resources, sum is set to TOT for all modes. PASS2 continues.
***SYSGEN DEFAULTED ONE TAPE GDEF RESOURCE TO 1.	At least one GDEF resource must exist for boot-time requirements. Therefore PASS2 has overridden the user specification of zero.
***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.

Table 53. PASS2 Messages (cont.)

Message	Description
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.
***xx CANNOT BE ASSIGNED – ENTRY IGNORED	The logical device name xx is always automatically generated by the system and should not be specified on the :LDEV command. PASS2 continues.
***xx DEVICE NOT DEFINED ***PASS2 UNABLE TO CONTINUE	Either CR or LP or both are missing on the :SDEVICE command. PASS2 aborts.
***xx DEVICE TYPE NOT SYMBIONT – ENTRY IGNORED	The device type xx was not specified as a symbiont device on a :SDEVICE command. PASS2 continues.
***xxxx LABEL MORE THAN 2 CHARACTERS. IT WILL BE IGNORED.	The user has specified an LDEV name that is greater than two characters in length. (The name specified is indicated in the xxxx field.)
***NAME A DEVICE TYPE – NAME IGNORED	The name cannot be a type mnemonic. PASS2 continues.
***NAME DUPLICATED – FIRST ONE USED	A logical device name was specified twice. PASS2 continues.
***xx SYMBIONT DEVICE TYPE NOT SPECIFIED – DEFAULT 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	
***A COMBINATION OF AN X560 AND MOS SPECIFICATION IS ILLEGAL	An X560 and MOS are incompatible. PASS2 aborts.

Table 53. PASS2 Messages (cont.)

Message	Description
***AVGSER IS OUT OF RANGE – DEFAULT (1) IS USED	Self-explanatory. PASS2 continues.
***COMBINATION OF SIG6/SIG9/X560 BIG-ILLEGAL OR ALL OF THESE ARE MISSING	See the description of the :MON command for a list of the legal combinations. PASS2 aborts.
CORE SPECIFICATION TOO LARGE FOR THIS SYSTEM. 128K SUBSTITUTED.	Only SIG9 or X560 together with BIG may exceed 128K.
***MINICOC OPTION WILL BE IGNORED SINCE TP HAS ALSO BEEN SPECIFIED	The TP and MINICOC options are mutually exclusive. PASS2 continues.
***NOT BOTH ANSDET AND ANSPROT – IGNORED	Only one of these options should be specified. Both are ignored. PASS2 continues.
***SINCE AN MC WAS SPECIFIED IN THE DEVICE COMMANDS, AN X560 SHOULD BE INCLUDED ON THE :MON CC.	Self-explanatory. PASS2 continues.
***SINCE AN X560 WAS SPECIFIED, AN MC DEVICE SHOULD HAVE BEEN INCLUDED IN THE DEVICE CONTROL COMMANDS	Self-explanatory. PASS2 continues.
***SYNTAX ERROR – 'X' EXPECTED	'X' is ')', ', ', or '('. PASS2 searches for the next opening parenthesis.
***TROUBLE WITH SPEC:HAND HANDLERS RECORD NOT INCLUDED	An I/O error has occurred with the SPEC:HAND file.
***UNRECOGNIZED DEVICE SYMBOL ON ONE OF THE DEVICE CARDS. SYMBOL IS: x	Check through the :DEVICE commands for the symbol specified as a channel definition. PASS2 aborts.
***AUTO AND NOAUTO BOTH SPECIFIED – AUTO USED	The NOAUTO option is ignored.
***AUTO ENTRIES EXCEED NCPU – EXCESS IGNORED	The value specified for NSCPU is less than a value specified on the AUTO option.
***n AUTO VALUE IS GREATER THAN 3. IT WILL BE IGNORED.	Self-explanatory.
***INTS ENTRIES EXCEED NSCPU – EXCESS IGNORED	The value specified for NSCPU is less than the number of values specified on the INTS option.
***INTS ENTRIES NOT EQUAL TO NSCPU – DELAYED ABORT	The number of values specified on the INTS option is less than the number of secondary CPUs. PASS2 aborts.
***MAXQ VALUE IS>32767 – DEFAULTED TO 32767	Self-explanatory.
***MAXQ VALUE IS LESS THAN CORRESPONDING MINQ VALUE. THEREFORE THIS MAXQ IS SET TO MINQ MATE.	Self-explanatory.

Table 53. PASS2 Messages (cont.)

Message	Description
***MAXQ VALUE IS ZERO - DEFAULTED TO 2000	Self-explanatory.
***MINQ VALUE IS ZERO - DEFAULTED TO 10	Self-explanatory.
***NO VALUE FOR NSCPU. THEREFORE A NON-MULTIPROCESSING SYSTEM WILL BE GENERATED.	Either NSCPU was not specified or was specified with a value of zero.
***NSCPU VALUE > 3. VALUE 3 HAS BEEN USED.	Self-explanatory.
COC Messages	
***A LINE WAS TAGGED AS A 2741 TERMINAL TYPE BUT WAS NOT DECLARED AS SUCH WITHIN A 2741 OPTION.	Terminal types and lines must be consistent. In this case, the running system treats the types and lines as non-2741s.
***A LINE WAS DECLARED AS A 2471 BUT WAS TAGGED AS A NON-2471 WITHIN A TERMINAL OPTION.	Terminal types and lines must be consistent. In this case, the running system treats the types and lines as 2741s.
***COCx 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' OR 'OUT' LOCATION NOT IN SAME GROUP AS PREVIOUS ONES	Self-explanatory. PASS2 discontinues processing of the :COC command.
***COCx LINES>64 – DEFAULT TAKEN	Self-explanatory. COC continues.
***COCx 'OUT' LOCATION NOT 'IN' LOCATION+1	Each COC input interrupt should be one higher in priority that the corresponding output interrupt. 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.
***ERROR IN COCDIO SPECIFICATION. IT MUST BE GREATER THAN PREVIOUS. PASS2 OVERRIDE USED.	A COCDIO specification is in conflict with a previous COCDIO address. COCDIO addresses must be generated in increasing order. PASS2 will ignore specification and generate an address of one greater than previous COC address.
***NOT ABLE TO GET WORK PAGES TO READ SPEC :HAND	No dynamic core pages were available at the point where the SPEC :HAND file was to be read into core to be updated. PASS2 continues.

Table 53. PASS2 Messages (cont.)

Message	Description
***PASS2 ABORT. EITHER A DEVICE OPTION HAS BEEN OMITTED OR 1 TOO MANY COC OPTIONS WERE GIVEN.	The :COC is considered the first COC. There must be a device option included in the options following :COC and before the next COC specification. Also, a device option must follow any subsequent COCs.
***TROUBLE WITH SPEC :HAND - TRANSLATE TABLES LOST	The type of I/O error is listed after this message and PASS2 aborts.
***TYPE > 7 INVALID - DEFAULTS USED	Self-explanatory. COC continues.
***WARNING :BUFFERS < 3* LINES	The total number of buffers was less than three times the total number of lines. COC continues.
IMC Messages	
***AIRM MUST BE LESS THAN OR EQUAL TO RASIZE. AIRM HAS BEEN SET EQUAL TO RASIZE.	Self-explanatory. PASS2 continues.
***AIRTO IS GREATER THAN 500000. AIRTO HAS BEEN SET TO 500000.	Self-explanatory. PASS2 continues.
***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.
***MAXB > # OF PARTITIONS. DEFAULTED TO # OF PARTITIONS.	Self-explanatory. PASS2 continues.
***MAXEXPIRE < EXPIRE - EXPIRE VALUE USED	Self-explanatory. IMC continues.
***MAXG < 10 - DEFAULT (10) USED	Self-explanatory. IMC continues.
***MAXG > 255 – DEFAULT (8) USED	Self-explanatory. IMC continues.
***MINTIME > QUANTA – QUANTA VALUE USED	Self-explanatory. IMC continues.
***OPRIO < X'C0' OR > X'FF'. DEFAULT USED (X'FC')	Self-explanatory. PASS2 continues.
***RAMAX MUST BE LESS THAN OR EQUAL TO RASIZE. RAMAX HAS BEEN SET TO RASIZE VALUE.	Self-explanatory. PASS2 continues.
***RASIZE NOT SPECIFIED. READ AHEAD TABLES NOT GENERATED.	Self-explanatory. PASS2 continues.
***RATO OUTSIDE LEGAL RANGE OF 0-32767, DEFAULT (10000) USED	Self-explanatory. (The default is 10000.) PASS2 continues.

Table 53. PASS2 Messages (cont.)

Message	Description
***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 VALUE 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.
SPROCS Messages	
***ATTEMPT TO REDEFINE xxxxxx. IGNORED	An attempt was made to redefine a standard processor without REPLACE. PASS2 continues.
***ENQ AND RTNRRT IGNORED. PASS2 SUPPLIES CORRECT OVERLAY NAMES.	ENQ and RTNRRT are the former names of ENQOV and RTOV. These overlays are automatically supplied by SYSGEN when appropriate. Therefore they need not be specified on the :SPROCS command and will be ignored if present.
***ENQ AND RTNRRT IGNORED. PASS2 SUPPLIES CORRECT OVERLAY NAMES.	ENQOV (which is equivalent to ENQ) and RTOV (which is equivalent to RTNRRT) are supplied by PASS2 when appropriate. Therefore, user requests for these monitor overlays are ignored.
***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.
***INSUFFICIENT SPACE – SPROCS ABORTED	Too many overlays have been selected for the available core space to generate M:SPROCS.
***MOSPACE OPTION NOT VALID. IT WILL BE IGNORED.	MOSPACE is no longer a valid option.
***xxxxxxx NOT STANDARD PROCESSOR. CAN'T REPLACE.	An attempt was made to replace a nonstandard processor. PASS2 continues.
***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 of granules per physical cylinder. It is set to an appropriate value obtained from S:CYLSZ (64 for the 7242/7270 or 110 for the 7260/7275).
***TROUBLE WITH SPEC:HAND FILE. NO MONITOR OVERLAY NAMES RECORD GENERATED.	An I/O error occurred when opening or writing this file. PASS2 continues but PASS3 for this system will abort if allowed to run.

Table 53. PASS2 Messages (cont.)

Message	Description
FRGD Messages	
***ADDRESS< 10000. DEFAULTED TO 10000.	The address supplied for the RESDF option is< 10000.
***NO CORE FOR FRGD, INTLB IGNORED	Self-explanatory. PASS2 continues.
***LABEL HAS ILLEGAL VALUE. IGNORED.	The value falls outside of the three permissible ranges. FRGD continues.
***INTLB WITHOUT FRGD. OPTION IGNORED.	Self-explanatory. PASS2 continues.
***SIZE OUT OF RANGE. DEFAULTED TO 10.	The RESDF size specification must be in the range 0-999.
***TROUBLE WITH SPEC:HAND – RTROOT NOT INCLUDED	I/O error in opening, reading, or writing SPEC:HAND file.
XLIMIT Messages	
***xxxx xxxxxx DEFAULT>MAX – DEFAULTS SET TO MAX	This message applies to :BLIMIT, :OLIMIT, and :GLIMIT commands and is self-explanatory. PASS2 continues.
***DEFAULT/MAX VALUES MAY NOT BE > 32767. SYSGEN WILL DEFAULT TO 32767.	The default or maximum value specified is greater than 32767.
***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.
***SIZE OUT OF RANGE. DEFAULTED TO 0.	The size value for the REFDEF option is out of range.
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 ISSUED	The maximum and minimum values are valid. 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.

Table 53. PASS2 Messages (cont.)

Message	Description
***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 *xxxx* 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.
GHOST Messages	
*** I/O ERR XX WHILE WRITING M:GHOST FILE	An I/O error occurred while trying to write the M:GHOST file. The error code is specified by xx.
***PRIORITY VALUES MUST LIE BETWEEN C0 AND FE. VALUES OUTSIDE THE RANGE ARE DEFAULTED TO C0.	A user-specified priority is outside the range of C0-FE.

LOCCT PROCESSOR

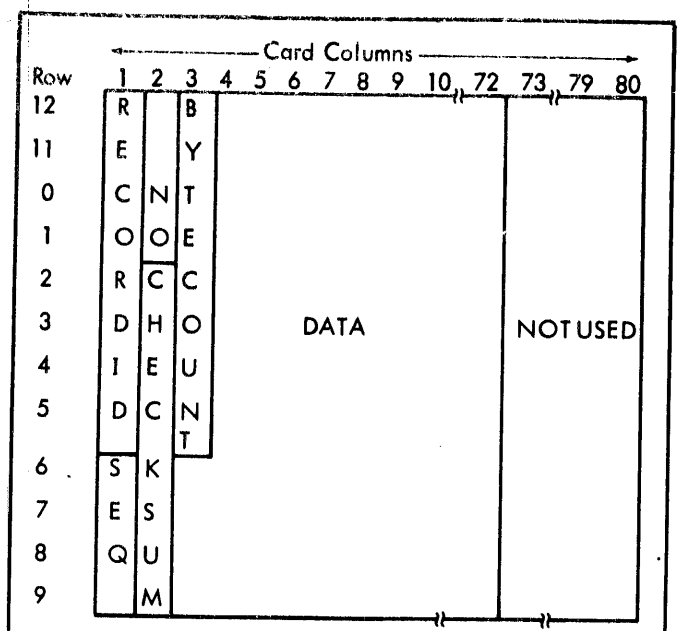
The LOCCT processor provides an optional phase of system generation that generates a file defining the elements and load structure of a user's processor or monitor. LOCCT may be called during system generation to create, in the current account, a permanent disk file containing the LOCCT, ROM, and TREE 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 43.

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.

†This control command replaces the !LOAD command for this type of process and contains the same information the !LOAD command would normally contain.



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 43. LOCCT Record Format

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 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 54 contains a list of error messages for LOCCT. All of these messages are output on the LL device.

(This page intentionally left blank.)

Table 54. 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 processor, a library subroutine, or the monitor (e.g., X, 9EDIT, FMGE). The id must not be longer than ten characters.

option is one of the optional parameters (SAVE or DELETE) described 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).

[†]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 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 as its overlay. In the absence of a loader specified via the F:LOADER DCB assignment, PASS3 will link to LOADER in account :SYS. When the loader completes its function, it will do an M:LDTRC call on the monitor requesting a return to the calling program; 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 must be loaded by PASS3 to take advantage of PASS3's ability to form the HANDLERS file automatically and to permit PASS3 to remove optional overlays (determined by options included or omitted on the PASS2 control commands) from the LOCCT prior to its M:LINK call to the loader. 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. If desired, the user may assign F:LOADER to a file in an account other than :SYS. PASS3 will then do an M:LINK to the alternate loader. When an MPC device is specified in PASS2, PASS3 expects to do an M:LINK to a firmware loader called FIRMLDR. Normally FIRMLDR resides in :SYS. If desired, the user may assign F:FIRMLDR to the MPC firmware loader in an account other than :SYS.

RECOVER must also be loaded by PASS3. It must be loaded after its associated monitor so that PASS3 may determine the load bias for the RECOVER load module. This bias is calculated by adding four pages to the M:MON start address after rounding the address to the next highest page.

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 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.

Example:

```
!JOB JX,LOADAUTO,1
!ASSIGN M: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.

When the LOCCT table to be processed is for M:MON, PASS3 checks for any MPC devices specified during PASS2. If at least one MPC device was specified and the F:FIRMLDR DCB is not set to DEVICE=NO, PASS3 does an M:LINK to the firmware loader. When complete, the firmware loader returns to PASS3; PASS3 continues processing M:MON.

Example:

```
!JOB :SYSGEN,LOAD M:MON,1
!ASSIGN F:LOADER,(FILE,LOADER,ACCLDR)
!ASSIGN F:FIRMLDR,(FILE,FIRMLDR,ACCMPC)
:M:MON
:Next control command
```

This example will obtain the LOCCT table for LOCCT M:MON from account :SYSGEN and will load M:MON via the loader in account ACCLDR. PASS3 will link to the firmware loader in account ACCMPC provided an MPC device was specified during PASS2, which generated the SYSGEN modules for M:MON.

For a monitor, 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.

PASS3 next attempts to read the HANDLERS2 record in the SPEC:HAND file. If the read is unsuccessful, only a HANDLERS and ROOTHAND file are generated. PASS3 reads INITRCVR and SSDATU files and copies them to the file it creates under the name of ROOTHAND. The BASHANDL (the basic handlers file) is then copied to the file named HANDLERS. PASS3 then reads the HANDLERS record in the SPEC:HAND file to obtain the names of the remaining handler files that need to be added to the HANDLERS file. Each unique file name (with the exception of those names shown in Table 55) causes PASS3 to open the corresponding file. When all names have been processed, PASS3 closes and saves the HANDLERS file. In the event that the HANDLERS record exists, PASS3 copies SSDATU and INITRCVR to the HANDLERS2 file and then copies the BASHANDL file to the HANDLERS2 file. It then reads the files named in the HANDLERS2 record of SPEC:HAND (with the exception of those names shown in Table 55), and adds

†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.

these files to the HANDLERS2 file. When all files have been copied, PASS3 reads the files named in the HANDLERS record in the same manner. The resulting files are copied into the HANDLERS file.

To summarize, the BASHANDL file is copied to either the HANDLERS file or the HANDLERS2 file depending on the absence or presence of the HANDLERS2 record. The file names given in these two records are copied to the file of the same name as the record with the exception of those names listed in Table 55.

Table 55. Handlers in BASHANDL File

Name	Device
KBTIO	TY
CRDIN	CR
PRTOUT	LP
PRTOUTL	LP
DISCIO	DC
DPAK	DP
DISKAB	DP

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., if 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 55).

PASS3 MESSAGES

Table 56 contains PASS3 messages. All of these messages are output on the LL device.

Table 56. 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 INVALID 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.

Table 56. PASS3 Messages (cont.)

Message	Description
**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 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.
***M:MON MUST BE A TREED LOAD MODULE. PASS3 WILL NOT ATTEMPT TO LOAD IT.	PASS3 continues to next control command.
***M:MON NOT SUCCESSFULLY LOADED	The Loader has indicated that there was a problem when trying to load M:MON. PASS3 exits.
***M:MON START ADDRESS = xxxx. RECOVER MODULE BIAS = yyyy.	xxxx is the value read from the M:MON load module in the current account. yyyy is the value calculated by PASS3 based on the value xxxx.
***MODULE NOT SUCCESSFULY LOADED	The Loader has indicated that there was a problem while trying to load this module. PASS3 may or may not exit depending on whether ALL or MON was specified.
**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 alphanumeric. PASS3 continues to the next name.
****OPEN/READ BASHANDL FILE ERR/ABN = xxxx	PASS3 could not obtain a given file when forming the HANDLERS or HANDLERS2 file. The value xxxx is the error or abnormal code. Trouble was encountered in the file being copied to the HANDLERS or HANDLERS2 file. PASS3 continues to the next command.
***OPEN/READ M:MON FILE ERR/ABN = xxxx. LOAD OF RECOVER NOT POSSIBLE.	PASS3 was unable to read the M:MON load module from the current account. Therefore, the RECOVER load module cannot be generated. The value xxxx is the error or abnormal code for the attempted read of M:MON.
***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.
**OPTION NOT 'MON' or 'ALL' - NONE ASSUMED	PASS3 will not abort if any load is unsuccessful. (Information only.)
PASS3 ABORTING. F:LOADER NAME MUST BE LESS THAN OR EQUAL TO 11 CHARS. IN LENGTH.	The F:LOADER DCB only has room for three words for the file name. PASS3 exits to the monitor.
PASS3 ABORTING. F:FIRMLDR NAME MUST BE LESS THAN OR EQUAL TO 11 CHARS. IN LENGTH.	The F:FIRMLDR DCB only has room for three words for the file name. PASS3 exits to the monitor.
####PASS3--COMPLETED####	PASS3 returned to the monitor.

Table 56. PASS3 Messages (cont.)

Message	Description
####F00PASS3--IN--CONTROL####	PASS3 has been entered.
***PROBLEM WITH LOCCT – NO MATCH WITH MISSING OVERLAY NAMES RECORD PRODUCED BY PASS2	This error can occur only when loading a monitor. No attempt is made to load the monitor and PASS3 continues to the next control command.
**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.
***THE FOLLOWING OVERLAYS HAVE BEEN REMOVED FROM THE M:MON LOCCT AS PER PASS2 REQUIREMENTS	Overlay names removed are listed for information purposes only.
***UNRECOGNIZED KEY IN SPEC:HAND = xxxxxxxx	The key found was not HANDLERS, HANDLERS2, or OVNAMES.

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 for 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 40). 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 57 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.

: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 started 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.

Table 57. Files Automatically INCLUDed

PO Tape (from current account)	BO Tape (from :SYS account)
BPM	XDELTA
UTS	LOGON
SIG7FDP	TEL
:BLIB	SUPER
M:CDCB	DEFCOM
M:OCDCB	SYMCON
M:BIDCB	ANLZ
M:CIDCB	ERRMSG
M:SIDCB	GHOST1
M:EIDCB	RECOVER
M:BODCB	M:MON
M:CODCB	ALLOCAT
M:SODCB	FIX
M:PODCB	CCI
M:GODCB	LOADER
M:LODCB	PASS2
M:DODCB	LOCCT
M:EODCB	PASS3
M:LLDCB	DEF
M:SLDCB	PCL
M:ALDCB	
M:LIDCB	
T:P2SI	

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. All random files will be ignored. 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. All random files will be ignored.

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,C00
!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 C00
:INCLUDE (FMGE)
:WRITE BO
:IGNORE (K13,K15)
:WRITE PO,C01
!Next monitor control command
```

DEF MESSAGES

Table 58 contains DEF messages. All messages are output to the LL device.

Table 58. DEF Messages

Message	Description
4300 CAN'T FIND A RECORD WITH THAT KEY. READING LMname PAGE.	M:MON, XDELTA, FIX, ALLOCAT, GHOST1, or RECOVER must not contain any RES greater than 512 words. User additions to these LMs (e.g., handlers) must observe this restriction.
****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.
**CC TYPE 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. NOT COPIED – RANDOM.	A random file was encountered amongst keyed (PO) and consecutive (BO) files and was ignored. DEF continues.
***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.
****M:MON TREE OUT OF ORDER	M:MON was loaded with overlay not in inverse alphanumeric order. DEF aborts.
**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.

Table 58. DEF Messages (cont.)

Message	Description
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.
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.
*****UNKNOWN TYPE---CPV---USED	The user has specified a monitor type other than CP. DEF continues.
****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 DEF processor is called on-line, it types:

```
TYPE IS?      $
```

The response should be ↵ or CP ↵ to specify CP-V. The version number must follow the type preceded by a comma.

If PASS3 is called on-line, it types:

```
OPTION(ALL,MON)
```

The response should be "ALL" or "MON" followed by a ↵ or just a ↵ for none.

If PASS2 is called, it types:

```
PASS2 AT YOUR SERVICE
```

No response is required. The message indicates that PASS2 has started processing the SI information.

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 ↵
!PASS2 ↵
 ....PASS2 AT YOUR SERVICE
$:CHAN ↵
;CHAN
    .
    .
    .
$END ↵
 .....END OF PASS2.....
!SET  M:SI  DC/P3SI
      (P3SI is a file of PASS3 commands.)
!SET  M:LL  LP ↵
      (Output is to the line printer.)
!PASS3 ↵
OPTIONS (MON,ALL)?$MON ↵
!SET  M:SI  0 ↵
      (Return SI to terminal.)
!SET  M:PO  9T#CPPO ↵
      (Assign device serial number.)
!DEF ↵
TYPE IS?  $ CP,C00 ↵
$:INCLUDE  (A1) ↵
$:WRITE  PO ↵
$END ↵
!
```

SYSGEN EXAMPLES

To simplify the system generation process, standard monitor systems and standard processors are predefined in files on the CP-V release tapes. All LOCCTs and PASS2, PASS3, and DEF jobs are also included. These files are described in the -11 release element.

APPENDIX A. OPERATIONAL LABELS

Table A-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 A-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 A-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 B. PHYSICAL DEVICE NAMES

A physical device name is indicated by yyndd.

where

yy specifies the type of device (see Table B-1).

n specifies the channel letter (see Table B-2).

dd specifies the device number (see Table B-3), in hexadecimal.

Table B-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
MO	Message mode communications equipment
MC	(Maintenance console) remote assist terminal

Table B-2. IOP Designation Codes

IOP Letter (n)	Unit Address
A	0
B	1
C	2
D	3
E	4
F	5
G	6
H	7

Table B-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).

Table B-4. Xerox 560 Cluster/Unit Matrix

UNIT # \ CLUSTER #	0	1	2	3	4	5	6	7
0	A	B	H	N	T	Z	5	*
1	$	C	I	O	U	0	6	*
2	#	D	J	P	V	1	7	*
3	@	E	K	Q	W	2	8	*
4	:	F	L	R	X	3	9	*
5	*	G	M	S	Y	4	␣	*
6	*	*	*	*	*	*	*	*
7	*	*	*	*	*	*	*	*

* Reserved.

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.

D

E

F

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.

G

H

I

J

K

L

M

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.

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.

T

U

V

W

X

Y

Reader Comment Form

We would appreciate your comments and suggestions for improving this publication

Publication No.	Rev. Letter	Title	Current Date

How did you use this publication?

☐ Learning ☐ Installing ☐ Sales
☐ Reference ☐ Maintaining ☐ Operating

Is the material presented effectively?

☐ Fully Covered ☐ Well Illustrated ☐ Well organized ☐ Clear

What is your overall rating of this publication?

☐ Very Good ☐ Fair ☐ Very Poor
☐ Good ☐ Poor

What is your occupation?

Your other comments may be entered here. Please be specific and give page, column, and line number references where applicable.

Your name & Return Address

Thank You For Your Interest. (fold & fasten as shown on back, no postage needed if mailed in U.S.A.)

CUT ALONG LINE

FOLD ALONG LINE

PLEASE FOLD AND TAPE–
NOTE: U. S. Postal Service will not deliver stapled forms

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 59153 LOS ANGELES,CA 90045

POSTAGE WILL BE PAID BY ADDRESSEE

HONEYWELL INFORMATION SYSTEMS
5250 W. CENTURY BOULEVARD
LOS ANGELES, CA 90045

ATTN: PROGRAMMING PUBLICATIONS

FOLD ALONG LINE

Honeywell

XEROX Publication Revision Sheet

November, 1976

CORRECTIONS TO XEROX CONTROL PROGRAM-FIVE (CP-V)/SYSTEM MANAGEMENT REFERENCE MANUAL
(Xerox 560 and Sigma 6/7/9 Computers)

PUBLICATION NO. 90 16 74H-1(11/76)

The attached pages contain changes for the E00 version of CP-V. Pages in the H edition of the manual that are to be replaced are: title page/ii, iii through vi, 1 through 4, 11/12, 47/48, 51/52, 57/58, 77/78, 81/82, 99 through 108, 111 through 114, 125/126, 141 through 150, 153 through 172, 175 through 182, 189/190.

These changes will be incorporated into the next edition of the manual.

Revision bars in the margins of replacement pages identify changes. Pages without the publication number 90 16 74H-1(11/76) at the bottom are included only as backup pages; revision bars appearing in such pages identify changes made in a previous revision.

20422
1478
Printed in U.S.A.

File No.: 1X13

XL21A, Rev. 0

XEROX

Publication Revision Sheet

September 1978

CORRECTIONS TO XEROX CONTROL PROGRAM-FIVE (CP-V)/SYSTEM MANAGEMENT REFERENCE MANUAL
(Xerox 560 and Sigma 5/6/7/9 Computers)

PUBLICATION NO. 90 16 74H-2(9/78)

The attached pages contain changes for the F00 version of CP-V. Pages in the H edition of the manual (11/75) combined with revision pages labeled 90 16 74H-1(11/76) that are to be replaced are: title page/ii, iii through vi, 23/24, 35/36, 57 through 60, 61/62, 67/68, 71/72, 75/76, 85 through 88, 103/104, 111/112, 125/126-138, 139 through 150, 153 through 156, 157 through 180, 181 through 186, 191 through 196. (Note: Section 9, pages 126 through 138, has been deleted.) New pages being added are: 60.1/60.2, 72.1/72.2, 156.1/156.2, and 180.1/180.2.

Revision bars in the margins of replacement pages identify changes. Pages without the publication number 90 16 74H-2(9/78) at the bottom are included only as backup pages; revision bars appearing on such pages identify changes made in a previous revision. Revision bars adjacent to page numbers indicate that material on the page has been reorganized without the content being changed.

21814
5C978
Printed in U.S.A.

File No.: 1X13

XL21B, Rev. 0

Honeywell Information Systems
In the U.S.A.: 200 Smith Street, MS 486, Waltham, Massachusetts 02154
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario M2J 1W5
In Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F.

22275, 2C1178, Printed in U.S.A.

XL21, Rev. 0