

Type III Class A Program

IBM

**Control Program-67/Cambridge Monitor System
(CP-67/CMS) Version 3.1
Program Number 360D-05.2.005
System Description Manual**

Control Program-67/Cambridge Monitor System (CP-67/CMS) is a multi-access system which manages the resources of a System/360 Model 67 such that remote users appear to have a dedicated System/360 at their disposal. Within this 'virtual machine' the user may select the operating system of his choice, subject to certain restrictions noted in this manual. The Control Program (CP-67) component creates the time sharing environment in which many virtual 360s (users) can simultaneously access the system.

The Cambridge Monitor System (CMS) component is a conversational operating system, used from a virtual machine, which provides a comprehensive, easy-to-use set of programs (commands) which give the CMS user a wide variety of functions, including the ability to create additional commands or subsystems to satisfy his special requirements.

This manual provides an overview of the features available in the CP-67/CMS System.

Preface

This manual contains a general description of the Control Program-67/Cambridge Monitor System. For more detailed information reference the following:

CP-67/CMS Hardware Maintainability Guide,
GH20-0858-1

CP-67/CMS Installation Guide, GH20-0857

CP-67/CMS Operator's Guide, GH20-0856

CP-67/CMS User's Guide, GH20-0859

CP-67 Program Logic Manual, GY20-0590

CMS Program Logic Manual, GY20-0591

CP-67: Operating Systems
in a Virtual Machine, GH20-1029

Third Edition (September 1971)

This edition, GH20-0802-2, is a major revision obsoleting GH20-0802-1. This manual also replaces and obsoletes the IBM Cambridge Scientific Center Report entitled An Introduction to CP-67/CMS (320-2032).

This Type III Program performs functions that may be fundamental to the operation and maintenance of a system.

It has not been subjected to formal test by IBM.

Until the program is reclassified, IBM will provide for it: (a) Central Programming Service, including design error correction and automatic distribution of corrections; and (b) FE Programming Service, including design error verification, APAR documentation and submission, and application of Program Temporary Fixes or development of an emergency bypass when required. IBM does not guarantee service results or represent or warrant that all errors will be corrected.

You are expected to make the final evaluation as to the usefulness of this program in your own environment.

THE FOREGOING IS IN LIEU OF ALL WARRANTIES EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

This edition applies to Version 3, Modification Level 1, of Control Program-67/Cambridge Monitor System (360D-05.2.005) and to all subsequent versions and modifications until otherwise indicated in new editions or Technical Newsletters.

Changes are continually made to the information herein. Therefore, before using this publication, consult the latest System/360 SRL Newsletter (GN20-0360) for the editions that are applicable and current.

Copies of this and other IBM publications can be obtained through IBM branch offices.

A form has been provided at the back of this publication for readers' comments. If this form has been removed, address comments to: IBM Corporation, Technical Publications Department, 1133 Westchester Avenue, White Plains, New York 10604.

CONTENTS

I.	INTRODUCTION.	1
II.	COMPONENTS OF THE SYSTEM.	1
III.	SYSTEM ENVIRONMENT.	2
IV.	THE CONTROL PROGRAM-67.	6
	Virtual I/O	7
	Console Functions	8
	I/O Error Handling.	9
	CP-67 Core Requirements	9
V.	THE CAMBRIDGE MONITOR SYSTEM.	9
	File Management under CMS	10
	CMS Commands.	14
	File Creation, Maintenance, & Manipulation	14
	Language Processors	14
	Execution Control	14
	Debugging Facilities	15
	Utilities	15
	Control Commands	16
	Library Facilities	16
	CMS Core Requirements	16
	CMS Batch Monitor	16
VI.	RUNNING OTHER SYSTEMS UNDER CP-67	17
VII.	PERFORMANCE OF CP-67	18
VIII.	CONTROL PROGRAM CONSOLE FUNCTIONS	18
IX.	CMS COMMANDS.	22
	APPENDIX.	29
	A. CP-67 Minimum Configuration.	29
	B. Terminals Supported by CP-67 As Operator's Console	30
	C. Transmission Control Units Supported By CP-67.	31
	D. Other Devices Supported by CP-67	32
	E. Devices Supported by CMS	33
	F. CP-67 Restrictions	34
	INDEX	38

Figures

1. Arrangement of mini-disks on a disk pack 3
2. Virtual machines with varying configurations
and operating systems running concurrently 5
3. Directory entry for a typical CMS Virtual Machine 6
4. CMS Structure and I/O Devices 13

I. INTRODUCTION

CP-67/CMS is a time-sharing system designed to provide each user with the machine configuration he needs for his particular application. The philosophy of "virtual machines" is to create the environment of a real computer and its associated I/O devices through simulation. Consequently, a virtual machine must have counterparts to all the components of the real machine it is simulating. For instance, a virtual S/360 consists of an operator's console (terminal), virtual memory, virtual CPU and virtual I/O devices. And, because of the dynamic address translation feature of the 360/67, each user's effective memory can be up to 16 million bytes of addressable core storage - larger than the real memory of the Model 67.

Each user's virtual System/360 is tailored to his specifications. When executing, it is indistinguishable to the user from a real System/360.

II. COMPONENTS OF THE SYSTEM

The CP-67/CMS time-sharing system consists of two independent components: the Control Program (CP-67 or CP for short) and the Cambridge Monitor System (CMS). The Control Program creates the time-sharing part of the system to allow many users to simultaneously access the computer. The Cambridge Monitor System provides the conversational part of the system to allow a user to monitor his work from a remote terminal.

Both components are independent of each other. CP-67 can be used on an appropriate configuration without CMS and CMS can be run on a properly configured System/360 as a single-user system without CP-67 (see Appendix A through E for appropriate configurations). If CP-67 is used without CMS, an operating system or systems must be chosen to provide the conversational or production aspect of the system as CP only provides the time-sharing capability.

CP-67 is capable of running multiple S/360 operating systems concurrently as long as they do not include any timing dependencies or dynamically modified channel programs. Dynamically modified channel programs are those which are changed between the time the Start Input Output (SIO) instruction is issued and the end of the I/O operation; i.e., changed by the channel program or the CPU. However, certain types of self-modifying channel programs can be translated, including those generated by the OS/360 Indexed Sequential Access Method (ISAM). For a comprehensive set of CP-67 restrictions, see Appendix F.

CP-67 is also capable of running System/360 operating

systems along with CMS in a multi-programming mode concurrent with its usual time-shared, multi-access operation. If the System/360 operating system contains telecommunication facilities, or remote job entry / remote job output support, CP will allow that system to control the lines of the 2701, 2702, or 2703 transmission control unit and the user to DIAL into that system from a remote terminal.

III. SYSTEM ENVIRONMENT

The operator's console of a virtual machine is represented by the remote terminal that a person has used to access CP-67. This remote terminal represents both the on-line 1052 printer-keyboard and the entire console display of a S/360. By means of appropriate terminal commands, the switches and lights of the CPU console are simulated, thus providing the control of a virtual machine. These console commands constitute the basic command language of CP-67 and are called console functions. There are three types of console functions. The first type simulates the control panel and includes such commands as DISPLAY, EXTERNAL, BEGIN and IPL. A second type is called extended console functions since they go beyond the simulation of the console, and include commands such as DETACH, MSG, QUERY, SET, and LINK. The third type provides the system operator with special control features such as ATTACH a real device to a given user, SHUTDOWN the system, ENABLE and DISABLE communications lines, and TERMINATE output on the unit-record devices.

For virtual memory CP-67 uses the dynamic address translation feature of the 360/67 to provide up to 16 million bytes of addressable core storage for each user (the full range of 24 bit addressing); consequently, the user's effective memory space can be larger than the real memory of the Model 67. For performance reasons, virtual memory is normally provided in sizes ranging from 256K to 1,024K but it can be defined in any multiple of 8K with 8K also being the minimum size. The size of virtual memory is defined in the user's directory and can differ for each virtual machine. Virtual memory is completely private to each machine, although certain areas of virtual memory, such as in part of the CMS nucleus, can be shared among different users.

The virtual CPU is the normal CPU of the 360 Model 67 shared by the different virtual 360's using time-slicing techniques. CP-67 normally provides a CPU similar to a System 360 Model 65, but a special mode is available to provide the facility of a virtual model 67 CPU. Any System 360 instruction can be executed by the virtual machine except DIAGNOSE, which is used by the virtual machine to communicate with CP.

Virtual I/O devices are devices controlled by the virtual 360 and not by CP-67; consequently, the software which

supports the virtual I/O device must be present in the operating system running in the virtual machine. An I/O device may exist on the virtual 360 with a different address than that existing on the real Model 67. It can also be defined as having a size not available in the real world, such as a 7 cylinder 2314. It may be a subset, such as a subset of the lines on a 2702 or 2703 transmission control unit. There may be a virtual 360 which only contains four 2702 lines whereas the real machine has a 2703 with 88 lines. Similarly, a 2314 disk pack may be suballocated into "mini-disks" such as several 10 cylinder 2314's. In this way, many mini-disks can reside on the same physical disk pack allowing many more virtual 360's to be run at a given time than there are real disk drives available. Figure 1 portrays the arrangement of mini-disks on a disk pack, along with temporary space provided for use by the control program. It is generally true, however, that the type of virtual device (or program-compatible equivalent) must exist on the physical machine before it can exist on a CP-67 virtual machine. A major exception to this is simulating a 2311 on a 2314.

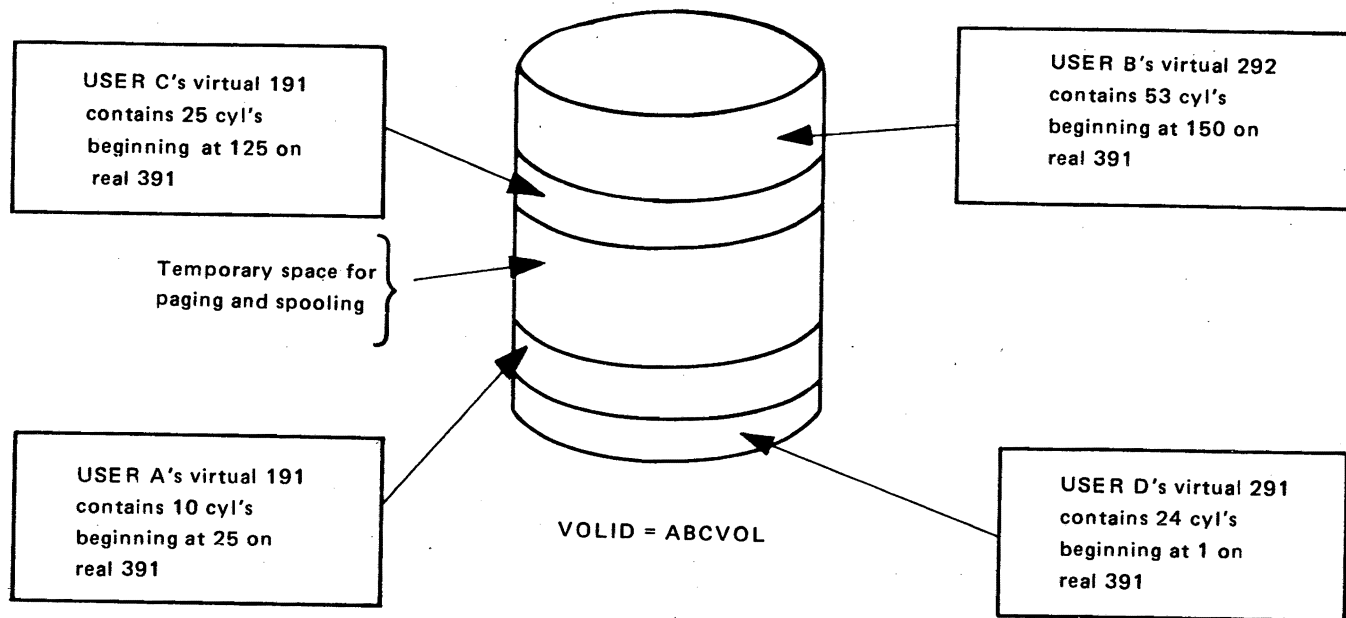


Figure 1

CP-67 builds and maintains for each user a virtual System/360 machine from a predescribed configuration. The virtual 360 is indistinguishable to the user and his programs from a real System/360, but it is really one of many that CP-67 is managing. CP allocates the resources of the real machine to each virtual machine, in turn, for a short "slice" of time, then moves on to the next virtual machine - thus, time-sharing.

Since the virtual machines are simulated, their configurations may differ from each other and from the real machine. For instance, the real machine may have 512K and eight disk drives and the virtual machine can have 768K and two disk drives. One virtual machine may have a virtual 2702 and run an OS teleprocessing system and another virtual machine that does not have a virtual 2702 may run CMS. One virtual machine may have a remote printer and a remote card punch while another virtual 360 may have a dedicated printer and 2250. Regardless of the configuration, each user controls his virtual machine from his remote terminal, which is, effectively, his operator's console.

Like real machines, virtual machines will operate most efficiently under an operating system. The Cambridge Monitor System (CMS) is designed to allow full use of a System/360 through a simple command language entered at the console (in the case of a CP-67 virtual machine, at the remote terminal). CMS gives the user a full range of capabilities -- creating and managing files, compiling and executing problem programs, and debugging -- using only his remote terminal. Since each user has his own virtual machine with his own copy of CMS residing "in it", he will not affect any other user; if he destroys the non-shared portions of the CMS nucleus or abandons the CMS system, he can re-IPL his virtual machine and continue without disturbing other users. CP-67 will prevent destruction of shared portions of virtual memory. In addition, since users cannot address memory "outside" of their virtual machines, CP-67 is protected from normal user error.

CMS also provides a batch version intended primarily for compile, load, and go type jobs coming from tape or cards. The batch monitor can be run from a virtual machine as a background system with conversational CMS users on other virtual machines.

Figure 2, on the following page, illustrates the virtual machine concept in CP-67 where virtual machines can have varying configurations and run different operating systems concurrently.

supports the virtual I/O device must be present in the operating system running in the virtual machine. An I/O device may exist on the virtual 360 with a different address than that existing on the real Model 67. It can also be defined as having a size not available in the real world, such as a 7 cylinder 2314. It may be a subset, such as a subset of the lines on a 2702 or 2703 transmission control unit. There may be a virtual 360 which only contains four 2702 lines whereas the real machine has a 2703 with 88 lines. Similarly, a 2314 disk pack may be suballocated into "mini-disks" such as several 10 cylinder 2314's. In this way, many mini-disks can reside on the same physical disk pack allowing many more virtual 360's to be run at a given time than there are real disk drives available. Figure 1 portrays the arrangement of mini-disks on a disk pack, along with temporary space provided for use by the control program. It is generally true, however, that the type of virtual device (or program-compatible equivalent) must exist on the physical machine before it can exist on a CP-67 virtual machine. A major exception to this is simulating a 2311 on a 2314.

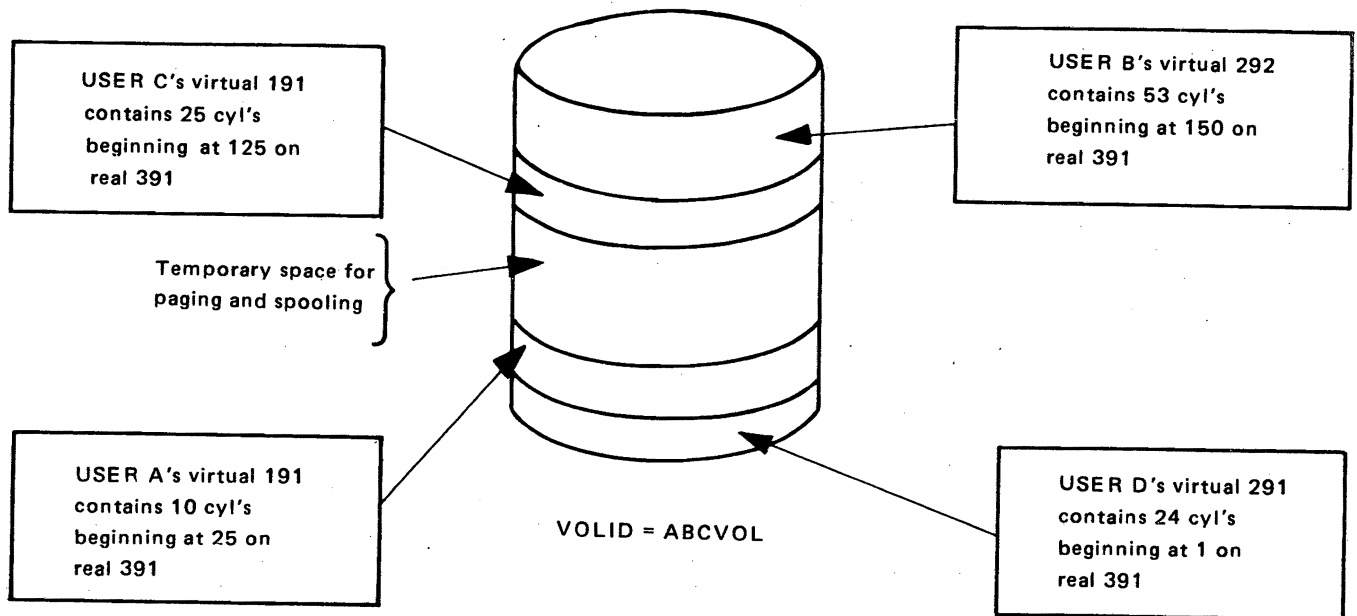


Figure 1

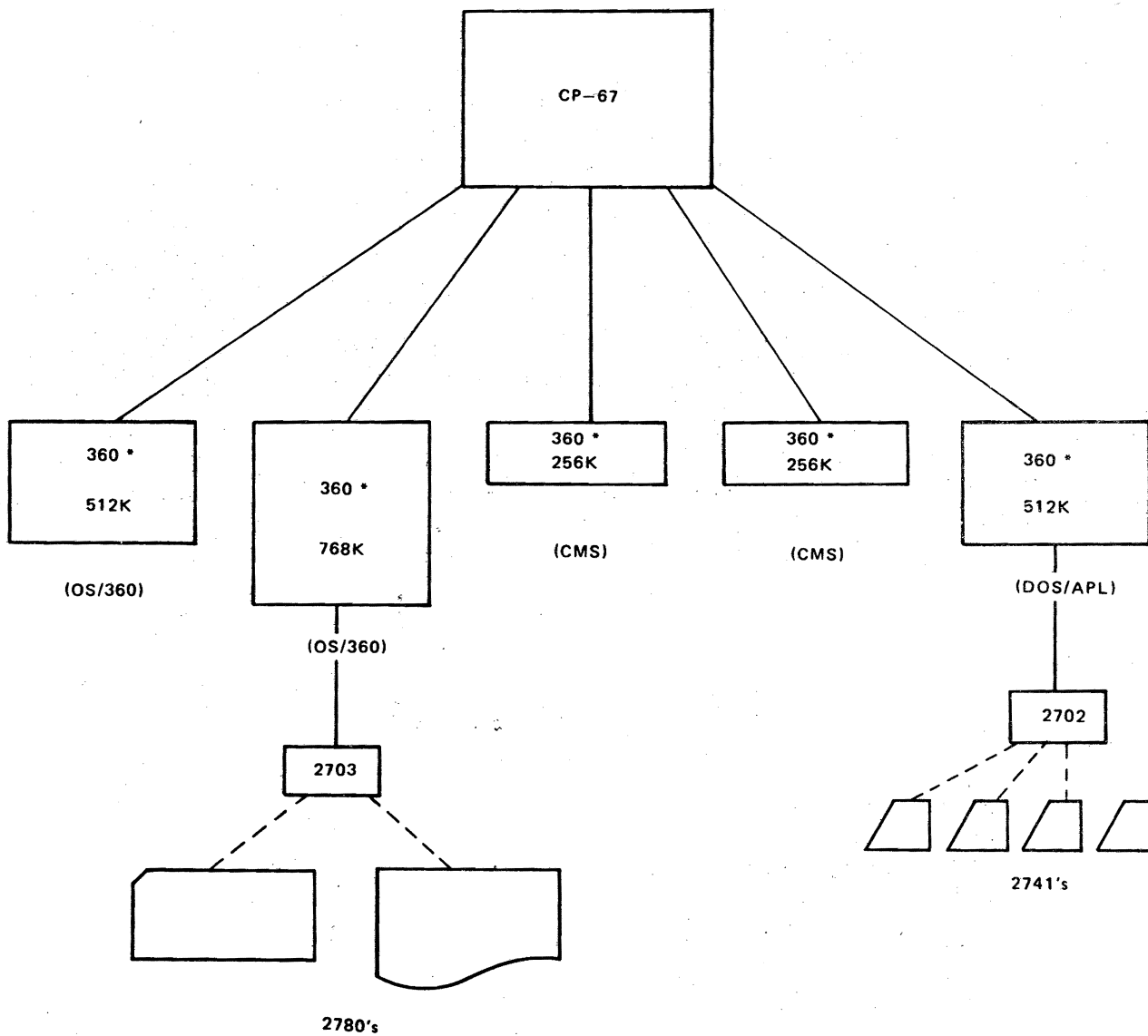


Figure 2

IV. THE CONTROL PROGRAM - 67

Before a user is authorized to use CP, he must be assigned (1) a USERID, which identifies him to the system, and (2) a password, which is checked when he "logs in". Associated with each USERID is information concerning accounting, privilege class, priority for CP dispatching purposes, options desired, and a table describing the virtual machine assigned to that user. Whenever he logs in, CP sets up this virtual 360 machine for him. Although all the virtual machines may be different, in most accounts they will be set up with at least the configuration expected by CMS. They will include at least 256K of virtual core storage, a minimum of two disk drives, a console (the terminal), a card-read-punch unit, and a printer. The real system will usually have a larger number of disk drives, a drum, tape drives, and perhaps more core storage. Figure 3 represents one description of a virtual machine. This user has 256K core, a 1052 online console with address 009, a card reader with address 00C, a card punch with address 00D, a printer with address 00E, a pseudo-timer device with address 00F, a read-only 54 cylinder 2314 with address 190, and a 10 cylinder mini-disk located on a 2311 labeled "ABCVOL".

Defining a Virtual 360

```
userid  USER  password,account,priv-class,priority,options
        CORE  256K
        UNIT  009,1052
        UNIT  00C,2540R
        UNIT  00D,2540P
        UNIT  00E,1403
        UNIT  00F,TIMR
        UNIT  190,2314,CMS190,000,054,RONLY
        UNIT  191,2311,ABCVOL,025,034
```

Figure 3. Directory Entry for a Typical CMS Virtual Machine

To expand the "available" core memory beyond the physical bounds of real core, a technique called "paging" is used by the system. Virtual core is divided into 4096-byte blocks of storage called "pages". The system keeps all but currently active pages on direct access secondary storage which is allocated only on demand; as active and inactive pages change status they are "paged" in and out of real core. While the paging operation is

being performed for one virtual machine, another virtual machine can be operating. Special hardware is provided on the System/360 Model 67 that translates, at execution time, the program's addresses into the current real addresses of the relocated pages. The paging operation, and resultant allocation of real core to a given user's pages is called "dynamic address translation" and is transparent to the user.

Only the Control Program (CP) may operate in the supervisor state on the real machine. All programs other than CP operate in the problem state. All user interrupts, including those caused by attempted privileged operations, are handled by CP, which then reflects to the user program only those the user program would expect from a real machine. The user's program will execute on the virtual machine in a manner identical to its execution on a real System/360.

VIRTUAL I/O

CP translates all virtual machine I/O operations into real machine I/O operations, in the following stages:

User

Issues START I/O

CP
translates

Virtual device addresses-->real device addresses
Virtual core storage addresses-->real core storage addresses

Other
users

executing

ensures required pages are in real core
builds channel command words string for user
issues SIO when channel is free
sets interrupt flag in user's virtual machine status table
returns control to user

User

Services reflected I/O interrupt

Virtual machine record I/O (card reader/punch/printer) are provided by a spooling mechanism.

A virtual machine can have dedicated unit record devices instead of sharing the system's unit record devices. With these dedicated devices, spooling is not performed by CP-67; the normal I/O requests are issued by the virtual machine, intercepted and translated by CP-67, and then the real I/O issued. In OS MVT, for example, double spooling will occur, one for OS and one for CP. The OS spooling operation may be bypassed with a request for a particular output device. CP-67 spooling may be circumvented by the ATTACHment of a real device to the virtual machine on which OS/360 is running.

Virtual I/O to the 1052 online operator's console is simulated, as CP must translate that I/O into sequences for the remote terminal from which the user has accessed CP-67 (either a 1050, 2741 (Correspondence or EBCDIC), or teletypewriter terminal compatible with the IBM Telegraph Terminal Control Type II Adapter (8-level ASCII code at 110 bps).

Terminals which are equivalent to those explicitly supported may also function satisfactorily; however, the customer is responsible for establishing equivalency. IBM assumes no responsibility for the impact that any changes to the IBM supplied products or programs may have on such terminals.

CONSOLE FUNCTIONS

The CP console functions allow the user to control his virtual machine from the terminal much as an operator controls a real machine. To perform an IPL, for instance, the user types "IPL" and a device address or the name of a "named" operating system, such as CMS. The user can stop his virtual machine at any time by depressing the ATTN key and request display of any portion of his storage and registers. He can modify the contents, if desired, and restart his machine. CP also recognizes a few special purpose commands, such as the XFER function mentioned above; the QUERY function to obtain the number of users on the system and their USERID's, as well as the number of current spooled files; the MSG function to communicate with other users; the DIAL function to connect the terminal to a multi-access system; and the ATTACH and DETACH functions to add or remove I/O devices from a virtual machine configuration (ATTACH can only be issued by a user with operator privileges, such as the system operator.) See Section VIII for a brief description of the CP-67 console functions.

CP-67 provides a warm start facility such that existing spooled input and output files are saved by CP for each user at system startup time. If an abnormal termination occurs, CP dumps

core storage to disk, unless the system operator has specifically directed all dumps to magnetic tape or a printer. If the core dump is to disk, CP automatically reloads itself and logs in the operator. The core dump is then available to a certain CMS virtual machine for printing under normal time-sharing operations. If the core dump is to tape or printer, the operator must bring the system up himself.

I/O ERROR HANDLING

CP also provides hardware serviceability aids. If a hardware error occurs, it is recorded in an area called SYSERR on the CP systems residence volume and an error message is sent to the CP operator. If an excessive number of errors occur on a particular device, a message stating the current error count and sense information for that device will also be printed on the CP operator's console. The SYSERR area can be accessed at any time by IBM Customer Engineers running programs provided under CMS which format and print out the recorded error information.

CP-67 CORE REQUIREMENTS

The CP system requires approximately 80K bytes of core storage. Additional core is assigned as CP free storage areas depending on the real core storage size. For example, for a 256K Model 67, 24K bytes of free storage are used; for each additional 256K of physical core storage, an additional 24K bytes of free storage areas are needed. Therefore, if the physical machine is 512K, 80K bytes are required by the CP system and an additional 48K bytes are required for work areas, or a total of approximately 128K. All core storage above that used for CP-67 residence and free storage areas is used for paging virtual machines, unless CP requires additional free storage, in which case it takes 4K bytes at a time.

V. THE CAMBRIDGE MONITOR SYSTEM

The Cambridge Monitor System (CMS) is a single-user, conversational operating system, capable of running on a real machine as well as on a virtual machine. It interprets a simple command language typed in at the operator's console (under CP-67, the user's remote terminal).

Whether running on a real (see Note below) or a virtual machine, CMS expects the following machine configuration:

CP-67/CMS System Description Manual

device	virtual address	symbolic name	
1052	009	CON1	console
2311,2314	190	DSK1	S(system)disk (read-only)
2311,2314	191**	DSK2	P(primary)disk (user files)
*2311,2314	192**	DSK3	T(temporary)disk (work space)
*2311,2314	000**	DSK4	A disk (user files)
*2311,2314	000**	DSK5	B disk (user files)
*2311,2314	19C**	DSK6	C disk (user files)
1403	00E	PRN1	line printer
2540	00C	RDR1	card reader
2540	00D	PCH1	card punch
*2400	180	TAP1	tape drive
*2400	181	TAP2	tape drive

at least 256K bytes of core storage, 360/40 and up

* The 2311 or 2314 for the temporary disk, the A, B and C disks, and the two 2400 tape drives are optional devices; they are not included in the minimum configuration.

** The reference between virtual address and I/O device may be changed at any time by the CMS LOGIN command.

Note: For use on a real machine not having this I/O configuration, the device addresses can be redefined at 'load' time.

Under CP these devices are mapped to different addresses and/or different devices and all console I/O is simulated. For instance, CMS expects a 1052 printer-keyboard operator's console, but most remote terminals are 2741's; CP handles all channel program modifications necessary for this simulation.

CMS allows the user to add his own programs for I/O devices not supported by the standard system. CMS also provides for dynamic specification of SVC routines.

FILE MANAGEMENT UNDER CMS:

Each CMS user is normally assigned two disks, one of which is shared with other CMS users, and up to four additional disks may also be assigned. Under CP-67 these disks are seldom complete disk packs. At the time a user is authorized to run on

CP-67/CMS, the size of each disk area is set by the system administrator, according to the needs of the user and the total amount of disk space available.

The shared disk (the system (SY) disk) contains the CMS nucleus which is loaded into the virtual machine by the IPL console function. Also on the disk are disk resident CMS commands, macro libraries, and program libraries. The disk is read-only; any attempt to modify the system files results in an error message.

The other five disks are known as the P, T, A, B, and C disks. These disks may be accessed for reading and writing or for only reading. They are allocated via the directory of CP-67 or by the user issuing the CP console function LINK. The user does not normally share the P and T disks with any other user, as they are usually accessible only to him on a read/write basis after he has logged in with the correct USERID and password. The P (primary) disk is used for files which are to be saved from one terminal session to the next. The optional T (temporary) disk, provides space for work files which need not be retained between sessions. This disk is normally erased whenever the user logs out of CP-67. The A, B, and C disks, which are also optional, are similar to the P disk; each can be accessed in a read/write mode by one CMS virtual machine. However, if any of the above disks (P, T, A, B, or C) are defined as read-shareable, they can be shared among CMS virtual machines on a read-only basis.

CMS uses a three-field file identification to catalog both system and user files. The fields are referred to as the filename, filetype, and filemode. Uniqueness of any one of the fields is sufficient to differentiate a file from other files. CMS maintains a "User File Directory" on each disk containing information on the file formats, sizes, and locations. Therefore the user may specify files by providing only the file identification or a portion of it.

All CMS disk files are written in 829-byte physical records. The system I/O routines handle placing of logical records into this format, allocating record blocks only as needed. To keep track of the files, CMS maintains chains of disk addresses linked to the User File Directory, which has an entry for each user file. The directory is brought into storage when the user logs in, and is updated on the disk at least as often as once per command if the status of the files has changed. This permanent copy is as current as possible, insuring an accurate directory if it is necessary to re-IPL CMS during a terminal session.

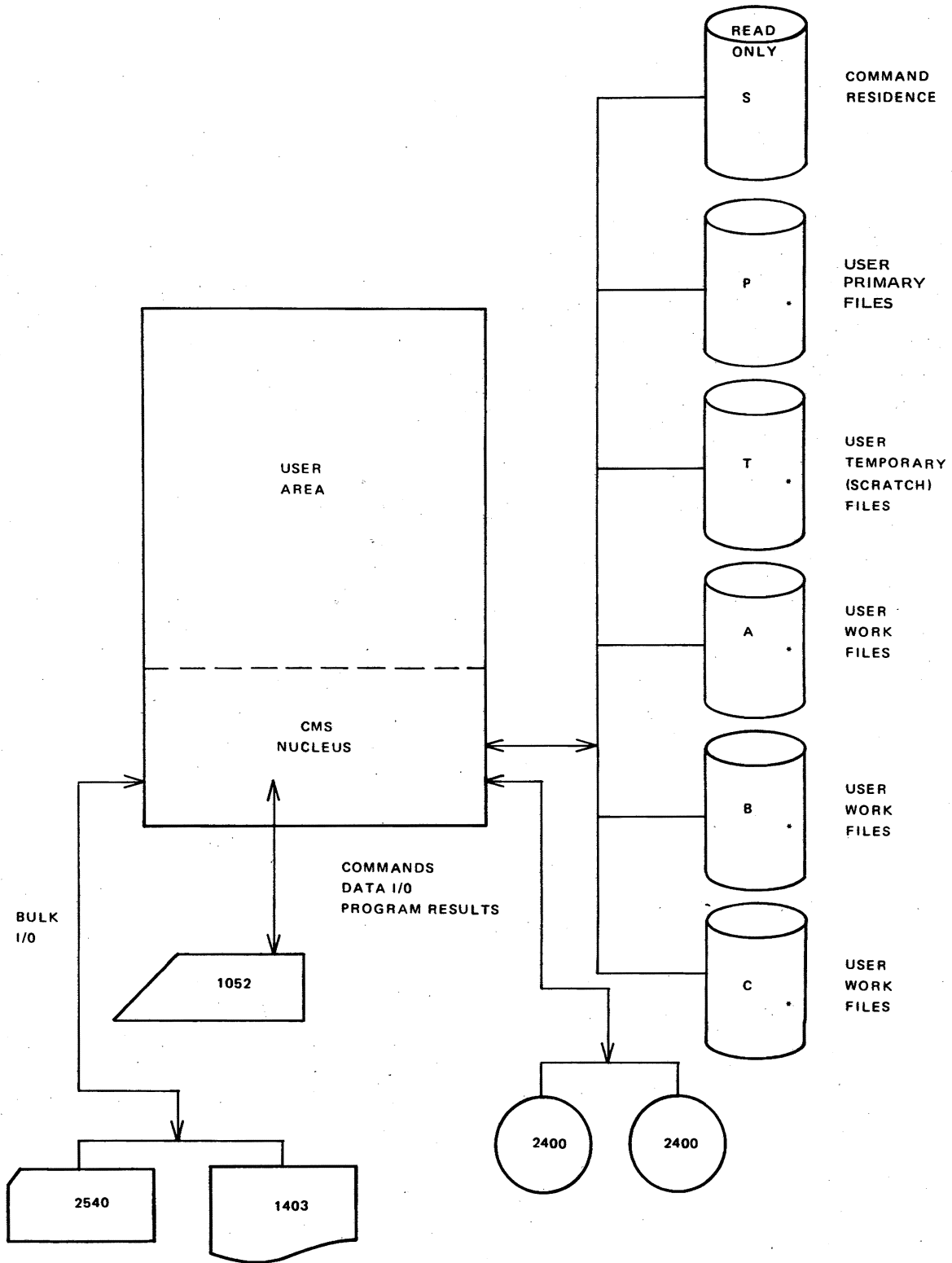
The directory is capable of cataloging up to 3500 separate files on a 203 cylinder 2314 disk pack. A single file can contain up to 12,848,000 bytes of data. In practice, the

user's disk will not normally require files of that length - the typical user needs less than 25 cylinders. Whenever CMS detects that only a few tracks are left on the user's disk, a warning message is typed, and the files currently open are closed. A program or command in execution is halted, so the user may create more free space on the disk by erasing some files, or by copying them from disk to other media.

Although most of the CMS commands operate on disk-resident files, the user also has access to the card-read-punch, printer, and tape drives. The commands, in general, create sequential files of fixed-length records; however, the programmer using the CMS I/O support routines is able to use any record format with either fixed-length or variable-length records. Fixed-length record files may also be read or written by direct access.

Files are automatically "opened" for reading or writing when the first read or write is issued. CMS routines automatically close files after every command and after user programs that complete normally.

Figure 4 illustrates the CMS I/O devices and their use.



* THIS CMS DISK IS NORMALLY A SUBSET OF A PHYSICAL DISK *

Figure 4. CMS Structure and I/O Devices

CMS COMMANDS:

A CMS command is (1) the name of a program residing in the nucleus or on any CMS disk, or (2) the name of a file containing other CMS commands. CMS commands fall into seven categories: file creation, maintenance, and manipulation; language processors; execution control; debugging facilities; utilities; control commands; and library facilities.

FILE CREATION, MAINTENANCE, AND MANIPULATION

The file handling commands of CMS allow the user to create and modify disk files via a context editor as well as to rename, copy, combine, split, update, erase, and print disk files. The user can also print or punch files on unit record equipment. He can create disk files from cards read via the card reader.

LANGUAGE PROCESSORS

The primary CMS language processors are the same ones used under Operating System/360 (OS). These include Assembler (F), FORTRAN IV (G), and PL/I (F). The Assembler produces object programs that may be executed under either CMS or OS, depending on the macros used in the source program. Special file handling routines for macro libraries are included. The FORTRAN and PL/I compilers also produce OS-compatible object programs. Diagnostics from the OS compilers are printed at the terminal unless suppressed by the user or directed to disk. Certain features of PL/I, such as teleprocessing, INDEXED, REGIONAL(2), and REGIONAL(3) file organizations are not supported by CMS at program execution time.

A text processor called SCRIPT is also included for text formatting. This document was produced by using SCRIPT. (See CMS SCRIPT User's Manual, GH20-0860.)

There are two additional language processors for use with CMS available as Type III programs from the IBM Program Information Department; they are SNOBOL, a string processing language, and BRUIN, an interpretive language. BRUIN, Brown University Interpreter, was adapted from the OS version of BRUIN developed at Brown University, Providence, Rhode Island. BRUIN provides two modes of operation: a desk calculator mode and a stored program mode.

EXECUTION CONTROL

The execution control commands allow the user to load his programs from single object decks called TEXT files (the filetype TEXT is reserved for relocatable object programs) or from a program library. He can pass a list of parameters to his

program from the terminal and specify the point at which execution is to begin. To bypass the relocating loader for each execution of the program, he can create a file consisting of an image of the portion of core storage containing his program and load that non-relocatable copy back at any time. Since the loading commands can be accessed by executing programs, overlay structures may be set up and dynamic loading can occur.

During program execution under CMS the user can fully interact with his program. By using the FILEDEF command the FORTRAN or PL/1 user can specify the characteristics of his data as well as the type of I/O device on which the data set resides. For example, input data to a FORTRAN or PL/1 program can be typed in at the terminal during program testing, and then be supplied from a disk file for production use.

The user also has the facility to create a file which is a series of commands, and then execute these commands by typing a single line; logic statements can be placed in the file with the commands such that if errors occur from a command, no more commands will execute from that file. This capability is called EXEC and allows a user to develop his own command language or sets of procedures. Other types of logic statements are the &GOTO, &IF, and &LOOP for logic control during execution. EXEC allows for the passing of variable arguments from the terminal as well as between EXEC files, as EXEC files can be nested and/or recursive.

DEBUGGING FACILITIES

The debugging command in CMS is called DEBUG. It allows the user to stop his programs at pre-determined points and examine his registers, PSW, and storage, and modify these if he so desires. This information may be typed out at his terminal or printed offline. A program interrupt gives control to DEBUG, as does the external interrupt caused by the EXTERNAL console function. The user may also employ the program tracing routines, which record all SVC transfers, or just those SVC's in which an error return is made.

UTILITIES

The utility functions in CMS provide for tape copying, disk file comparing, a disk file sort, and the dumping of files either by name onto the console or by cylinder locations onto the offline printer. There is a facility to dump files to tape and reload them onto disk. There are commands for converting files of fixed length records to variable length records, for converting BCDIC files to EBCDIC, and for obtaining statistics on file space.

CONTROL COMMANDS

There are other commands which give the user the facility to suppress and resume typing at the terminal and to kill program execution. One can also rename any command and define unique abbreviations.

The command VSET, gives the user the following capabilities: increased usage of the character redefinition functions: BLIP, CHARDEF, and LINEND, the facility to set the order of search for commands, to abbreviate the ready message, control the number of pages of core used for loader tables, and to control the release of user core pages. It is also possible to receive certain terminal responses in red type (provided the terminal is equipped with the appropriate red ribbon control feature).

LIBRARY FACILITIES

CMS provides library facilities for program libraries. The user can generate his own libraries or add to, delete and list entries in existing libraries. He can also specify which libraries to use for program assemblies as well as program execution.

Each of the CMS commands is described briefly in Section IX.

CMS CORE REQUIREMENTS

CMS requires 80K bytes of virtual memory for the nucleus, transient area and loader tables. The rest of core storage is available to user programs unless CMS requires additional free storage, in which case it is taken as needed.

CMS BATCH MONITOR:

As well as being a conversational monitor, CMS provides a batch facility for running CMS jobs. The CMS batch monitor accepts a job stream from a tape unit or the card-reader and writes the output either on tapes, the printer, or the card-punch. The job stream consists of CMS commands along with control cards and card decks for compile, load and go jobs for all the CMS supported compilers.

Just as the conversational CMS does, the batch monitor can run from either a virtual machine or a real machine. Under CP, it can be used as a background monitor along with other conversational CMS users.

To eliminate the possibility of one job modifying the

CMS batch monitor's nucleus in such a way as to affect the next job, the Batch monitor is re-IPLed before each job begins. Files can also be written onto the batch monitor's P-disk and then punched or printed, such as files written by Fortran programs; these files should be of limited size and considered as temporary, as they are erased at the completion of each job.

VI. RUNNING OTHER SYSTEMS UNDER CP-67

CP-67 allows multiple operating systems to be run concurrently in different virtual machines. One operating system might be a regular OS/360 batch system, either PCP, MFT, or MVT. The terminal from which the user logs in becomes the operator's console for that virtual machine and he runs his OS as he does on the real machine.

Modifications can be made to OS to allow multiple virtual machines to run OS from the same systems residence device, thus creating a read-only SYSRES volume. Changes can also be made to allow an OS user to have mini-disks, subsets of a real direct access device, such as 50 cylinders of a 2314, instead of using all 200 cylinders of that disk.

As well as running batch systems, CP-67 can run multi-access systems. The multi-access systems being run under CP can have high-speed and/or low-speed communications lines. In the case of low-speed lines, once the multi-access system has been loaded into a virtual machine and the communications lines prepared, users at remote terminals can become connected to this virtual machine by issuing the CP console function DIAL. Once the remote terminal is connected to that system, it is treated by CP-67 as an attached transmission line. The existence of a terminal is then known only by that virtual machine, and thus CP-67 console functions are not available. Only when that multi-access system disables the communications line will CP regain normal control of the terminal.

Some of the multi-access systems that have run under CP-67 are APL/360, DOS/POWER, HASP/RJE, and CPS. Other operating systems that have run under CP-67, in addition to CMS and the CMS Batch Monitor, are OS (PCP, MFT, MFTII, MVT, HASP, ASP), DOS, PS44, and many of the FE Diagnostics.

Note that although these operating systems have been run successfully under CP-67, particular applications or access methods may violate the timing and I/O restrictions described earlier and cannot be used. Each installation is responsible for making the final evaluation as to whether an application or program will operate properly under CP-67.

VII. PERFORMANCE OF CP-67

The number of virtual machines supported by CP-67 must be considered with respect to the type of operating systems being run in each virtual machine, as different operating systems put different loads on CP-67. One sample PL/I jobstream was run through OS/MVT with two initiators under CP-67 in 51 minutes (no other users), whereas the same jobstream was run through OS/PCP (using CP's spooling feature to overlap printing time) along with 25 other users in 40 minutes. The same PL/I jobstream was then run in two OS/PCP virtual machines simultaneously (no other users) in 37 minutes, thereby performing twice the workload in less time than one jobstream with OS/MVT. Thus, in cases where jobstreams can be split between different virtual machines, running multiple OS systems under CP-67 can sometimes provide more OS throughput.

In analyzing the performance of any operating system under CP-67, the workload being performed by all virtual machines must be considered. If a single virtual machine is run with an increase in elapsed time, that increased time must be weighed against the work being done by other virtual machines.

CMS was designed to run under CP-67, and to communicate with a single user at the console; it knows nothing about multiprogramming or multiple users. In consequence, CMS does not place a very heavy load on CP-67. Up to 40 CMS virtual machines have been run on a 512K Model 67 with a reasonable response time, for instance, three seconds to process an EDIT request.

CP-67 will add additional overhead to an operating system running under it, thus increasing the elapsed time in comparison to elapsed time in a stand-alone environment. The primary source of overhead for a single virtual machine is in the I/O processing function. As far as degradation is concerned, an I/O bound task will probably suffer the most.

VIII. CONTROL PROGRAM CONSOLE FUNCTIONS

Each of the CP-67 console functions that can be issued by a user from the terminal is described below. The privileged operator commands are included in the list following this one.

BEGIN	begins execution at the specified address or, if no address is given, at the location at which execution was interrupted.
CLOSE	releases the spooling areas containing input from the card reader, unless the user has

requested via the "SET" command that his input be saved. CLOSE releases output files to the printer or card punch.

- DETACH removes the specified device from the user's virtual machine configuration.
- DIAL is used in place of LOGIN to connect a user's terminal with a virtual telecommunications operating system or a virtual time-sharing system.
- DISCONN allows a user to disable the terminal and leave his virtual machine running.
- DISPLAY types at the terminal the contents of the specified general purpose and floating point register(s), core location(s), or program status word. For virtual 360/67, DISPLAY will also type the contents of the control registers.
- DUMP prints the contents of the specified general purpose and floating point register(s), core location(s), or program status word on the offline printer. For virtual 360/67, DUMP will also print the contents of the control registers.
- EXTERNAL simulates an external interrupt to the virtual machine.
- IPL clears core and simulates the Initial Program Load sequence on the specified unit.
- IPLSAVE simulates the Initial Program Load sequence on the specified unit, but does not clear core first.
- LINK attaches directory-defined virtual disks dynamically.
- LOGOUT releases the user's virtual machine, including his temporary disk area, and closes any spooling files which have not been released.
- MSG types the specified message at the terminal of the person whose USERID is specified.
- PSWRESTART performs a virtual PSW RESTART.

CP-67/CMS System Description Manual

- PURGE** erases spooled input or output files by device.
- QUERY** types out either the number of logged in and dialed users, the names of logged in users, the number of current spooled input and output files, CPU time used during the session, or the user's current virtual machine configuration.
- READY** simulates a device end for the specified unit.
- RESET** simulates the system reset key on the 360 console.
- SET** controls the saving of virtual card-reader files and the typing of messages at the terminal; SET TRACE allows tracing of interrupts and instructions; ADSTOP specifies a virtual instruction address at which to stop execution; APLBALL readys the system for APL input and output; ATTN controls whether the ATTN interrupt takes the user into the console function mode or goes directly to the virtual machine. LINEDIT allows the user to edit virtual machine 1052 input using the CP-67 @ and % line editing characters.
- SLEEP** invokes a special console function mode to facilitate the receiving of messages; the virtual machine is in a dormant state.
- SPOOL** directs spooled output and controls the reading of spooled input.
- STORE** replaces the contents of the specified general purpose and floating point register(s), core location(s), or program status word with the specified information. For virtual 360/67, STORE will also replace the contents of control registers 0, 2, 4 and 6.
- XFER** establishes a data path between a spooled output device and a spooled input device.

Each of the CP-67 privileged operator commands is described below.

ACNT	punches and resets accounting information.
ATTACH	adds I/O devices to a virtual 360.
DETACH	(rccu) removes real I/O devices from a virtual 360.
DCP	displays real core.
DIRECT	controls access to the system directory.
DISABLE	disables 2702 or 2703 lines for access to the system.
DMCP	dumps contents of CP-67 memory.
DRAIN	stops the output device when it reaches the end of the current spooled output file.
D_U_M_P	causes system ABEND via SVC 0 with subsequent system dump.
ENABLE	enables 2702 or 2703 lines.
KILL	removes an unwanted user from the system.
LOCK	makes parts of a user's virtual memory core resident and not available for paging.
MSG	sends message to certain or all users.
QUERY	types information regarding number of users on system, specific devices and terminals in use, user dumps, and core size of particular or all devices.
REPEAT	prints or punches up to 10 times the current spooled output file.
SET	sets various parameters in the system, such as date, time of day, and log message; allows tracing of all successful branches and all instructions.

CP-67/CMS System Description Manual

SHUTDOWN brings the system to an orderly stop.

SPACE single spaces the output of the printer.

START initiates output on a "DRAIN"ed device.

STCP stores into real core.

TERM terminates output on the printer or punch.

UNLOCK makes LOCKed parts of a user's virtual memory available for paging.

WNG sends a warning to a user.

IX. CMS COMMANDS

Each of the commands that can be issued by a user to CMS are described below.

ALTER changes all or part of the identifier (filename, filetype, and filemode) of a file stored on the user's disk without altering the contents of the file.

ASSEMBLE converts assembler language source code into relocatable object code using the OS/360 F level Assembler.

BLIP causes a specified string of characters to be typed at the console every 2 seconds of CPU execution. (See VSET)

BRUIN invokes the Brown University Interpreter, in which the user has a desk calculator mode and a stored program mode. BRUIN is available as a Type III program.

CHARDEF redefines the character delete symbol, the line delete symbol, the logical tab character, and the logical backspace character, which default to the @, @, #, and % respectively. CHARDEF also specifies the type of translation being defined by the characters represented, using IN, OU and IO. (See VSET)

CLOSIO signals the Control Program that I/O to offline unit record equipment has been completed and that the spooling areas for

this I/O may be processed. CLOSIO is generally issued automatically by the commands which access unit record equipment.

- CLROVER clears overrides set by the SETERR and/or SETOVER commands and causes all recorded trace information to be printed on the offline printer.
- COMBINE joins two or more disk files into a single file, concatenating them in the order given, moves files between disks, and changes file designations.
- COMPARE compares two disk files.
- CPFUNCTN issues CP console functions without leaving CMS.
- CNVT26 converts BCDIC files to EBCDIC.
- CVTFV converts files of fixed length records to variable length records.
- DEBUG allows the user to stop and re-start programs at specified points and to inspect and change the contents of registers, core locations, and hardware control words online.
- DISK causes a CMS disk file to be punched onto or read from cards which are in a special format.
- DUMPD prints the contents of a direct access record, specified by a CCHRR address, in hexadecimal on the printer.
- DUMPF types the contents of all or part of a specified file in hexadecimal on the console.
- DUMPREST dumps the contents of an entire disk to magnetic tape or restores the contents of an entire disk from magnetic tape.
- ECHO tests terminal line transmission by repeating as typeout whatever is typed in by the user.
- EDIT allows the user to create files on disk, to make changes to existing files from his terminal, and to peruse the contents of files.

- ERASE** deletes the entry for a specified file (or files) from the appropriate directory of a read-write disk, rendering the file inaccessible to the user, and freeing the disk area containing that file.
- EXEC** executes a file containing one or more CMS commands, plus logic statements, thus allowing a logical sequence of commands to be executed by issuing a single command.
- FILEDEF** allows the user to specify the I/O devices and certain data set characteristics which will be needed by his program. Also used to modify, delete and list current file definitions.
- FINIS** closes the specified file (or files) by writing the last record of that file on disk, updating the User File Directory, and removing the entry for that file from the user's table of active files.
- FORMAT** prepares a disk area for CMS use, or checks the number of cylinders and disk counts on an already formatted disk.
- FORTRAN** converts Fortran language source code into relocatable object code using the OS/360 FORTRAN G compiler.
- GENMOD** creates a non-relocatable core-image file on the user's permanent disk which is a copy of the contents of core between two given locations, thus creating a disk-resident command.
- GLOBAL** specifies (1) macro definition libraries to be searched during the assembly process or (2) text libraries to be searched when loading files containing relocatable object code.
- KE** controls the length of the line being typed at a terminal.
- KO** clears overrides during the currently executing command that were previously set by the SETOVER or SETERR commands and causes all trace information recorded by these commands to be printed on the offline printer.

- KT** stops typeout at the terminal for the duration of the currently executing command or user program.
- KX** terminates the currently executing program, updates the User File Directory, re-IPL's CMS, and returns control to the CMS command environment.
- LINEND** redefines the logical line-end character which allows multiple commands to be entered on one line. The default is the #. (see VSET)
- LISTF** either (1) types out at the terminal the identifiers, size, and creation date and time or change date and time of the specified disk file(s), or (2) creates a file on the user's permanent disk containing information for use by the EXEC and/or \$ commands. By specifying an asterisk for filemode, both read-only and read-write disks are examined for function (1); if a filemode is not specified, only read-write disks are examined.
- LOAD** reads the specified TEXT file(s) -- containing relocatable object code -- from disk or a library, loads them into core, and establishes the proper linkages.
- LOADMOD** reads a MODULE file -- which is in non-relocatable core-image form -- from disk and loads it into core.
- LOGIN** causes the user's files on a specified disk to be either saved or deleted, as specified. If LOGIN is not issued, all existing files on the P-disk will be saved and a user profile invoked. LOGIN operands can be used to specify hexadecimal disk address, disk mode, disk as a read-only extension of another disk, and specific files whose directories are to be brought into core (if disk is read-only).
- LOGOUT** executes any CMS command specified as an operand, and logs out of CMS transferring control to the Control Program.
- MACLIB** generates, adds to, replaces, deletes, or compacts macros in a specified macro library, or types out the contents of the dictionary of that library.

CP-67/CMS System Description Manual

MAPPRT types or creates a file containing the load map associated with the CMS nucleus.

MODMAP types the load map associated with a core-image MODULE file on the console.

OFFLINE creates disk files from card input, prints a disk file on the offline printer, or punches a disk file on the offline punch.

OSTAPE creates disk files from unblocked card images on tape.

PLI converts PL/I language source code into relocatable object code using the OS/360 PL/I F compiler.

PRINTF types at the terminal the contents of all or part of a specified disk file.

RELEASE releases a disk from a user's virtual machine when he has finished using it. Can also include 'DETACH'. Used in conjunction with LOGIN and/or LINK.

REUSE reads the specified TEXT file(s) -- containing relocatable object code -- from disk and loads them into core, establishing linkages with previously loaded files and changing the default entry point of these files to that of the first file specified in the REUSE command.

RT restores typing at the terminal that was previously suppressed by KT.

SCRIPT types or prints out the contents of the specified file, formatting it as indicated by control words contained in the text.

SETERR sets error overrides which will cause trace information to be recorded for each SVC-called program which returns with an error code in general purpose register 15.

SETOVER sets normal and error overrides which will cause trace information to be recorded for all SVC-called programs -- both those which are executed normally and those which return an error code in general purpose register 15.

SNOBOL converts and executes a card-image file in Snobol source language. SNOBOL is available as a Type III program.

SORT uses the specified positions within each record to arrange a disk file in ascending order and write the sorted output into a new file.

SPLIT copies the specified portion of a card-image file and either creates a new file or appends it to a second specified card-image file.

START begins execution of the loaded program(s) at the specified or default entry point and passes the address of a string of user arguments to the program(s).

STAT types statistics regarding the amount of disk space used, including number of files, on read-write or (if * is specified) on all disks currently logged in. A ? following parameter list asks for brief information on the disk, specifically whether 2311 or 2314 is read-only or read-write.

STATE tests whether a file exists.

SYN allows the user to rename any command and/or specify abbreviations to be used.

TAPE writes the contents of CMS disk files of any type or size onto magnetic tape, or restores these files by writing them from tape onto disk.

TAPEIO provides direct control of a tape unit, such as writing a tape mark on magnetic tape, erasing a gap on the tape, or moving the tape.

TAPRINT copies LISTING files from tape to printer.

TPCOPY copies tape files.

TXTLIB either (1) generates, adds to, or deletes modules in a specified text library, (2) types out the contents of the dictionary for that library, or (3) creates a file containing a list of entry points and control section names contained in that library.

CP-67/CMS System Description Manual

- UPDATE** updates the specified disk file with a file containing control cards, where each control card indicates whether the information immediately following it is to be resequenced, inserted, replaced, or deleted.
- USE** reads the specified TEXT file(s) -- containing relocatable object code -- from disk and loads them into core, establishing linkages with previously loaded files.
- VSET** provides redefinition of the following symbols: character delete, logical line end, line delete, backspace and tab characters, and hexadecimal representations of characters not on terminal keyboard; VSET IMPEX modifies the order of search for commands, VSET RDYMSG abbreviates the ready message, VSET LDRTBLS controls the number of pages of core used for loader tables and VSET RELPAG controls the release of user pages. If the terminal is equipped with the appropriate red ribbon control feature, VSET REDTYPE causes certain error responses to appear in red. VSET BLIP sets the character chosen to notify the user of every two seconds of CPU time used. (See BLIP, CHARDEF and LINEND)
- WRTAPE** copies files of fixed-length records from disk to tape.
- \$** executes a file containing one or more CMS commands, or loads into core a file which is in either core-image form or relocatable object code and begins execution of that file.

APPENDIX A

CP-67 Minimum Configuration

2067-1 or 2067-2 Processing Unit

Recommended feature:

#4434 Floating Storage Addressing (Model 1 only)

2365 Processor Storage

2860 Selector Channel

2870 Multiplexer Channel

1052 Printer-Keyboard

1403 Printer

2540 Card Read Punch

Three 2311 Disk Storage Drives or 2314 Direct Access Storage Facility

(2 Disk Storage Modules minimum)

2400 Nine-Track Magnetic Tape Unit, 800 or 1600 bpi

2702 or 2703 Transmission Control or

2701 Data Adapter Unit

APPENDIX B

Terminals Supported By CP-67 As Operator's Console

1051/1052 Model 2 Data Communication System

Features and Specifications:

Data Set Attachment (#9114)

IBM Line Adapter (#4647)

Receive Interrupt (#6100 or RPQ E27428) required

Transmit Interrupt (#7900 or RPQ E26903) required

Text Time-out Suppression (#9698) required

1056 Card Reader

2741 (-1,-2) Communication Terminals

Features and Specifications:

Data Set Attachment (#9114)

Data Set Attachment (#9115)

IBM Line Adapter (#4635, #4647)

Dial-Up (#3255)

Receive Interrupt (#4708) required

Transmit Interrupt (#7900) or Transmit

Interrupt Control (RPQ E40681) required

Print Inhibit (#5501) desirable

Line control for teletypewriter terminals* compatible with the IBM Telegraph Terminal Control Type II Adapter (8-level ASCII code at 110 bps).

* The customer is responsible for terminal compatibility with this program. IBM assumes no responsibility for the impact that any changes to the IBM supplied products or programs may have on terminals provided by others.

APPENDIX C

Transmission Control Units Supported By CP-67

2701 Data Adapter Unit

<u>Terminals</u>	<u>2701 Adapter</u>
8-level ASCII, 110 bps*	7885

* The customer is responsible for terminal compatibility with this program. IBM assumes no responsibility for the impact that any changes to the IBM supplied products or programs may have on terminals provided by others.

2702 Transmission Control

<u>Terminals</u>	<u>Terminal Control Base</u>	<u>Terminal Control</u>	<u>Line Adapter</u>
2741s, 1050	9696 or 7935	4615, 9684, 8200**	3233
8-level ASCII 110 bps*	9697 or 7935	7912	3233

** Feature 8200 on the 2702 is equivalent to the 2741 Break feature #8055 and the Type 1 Break RPQ E46765 on the 2702.

2703 Transmission Control

<u>Terminals</u>	<u>Line Speed Option</u>	<u>Line Set</u>	<u>Terminal Control</u>	<u>Line Bases</u>
2741s, 1050	4878	3205/6	4619, 4696, 8200***	7505
8-level ASCII, 110 bps*	4877	3205/6	7905, 7912	7505

*** Feature 8200 on the 2703 is equivalent to the 2741 Break feature #8055 and the Type I Break RPQ E53715 on the 2703.

APPENDIX D

Other Devices Supported by CP-67

Additional Devices Utilized by CP-67

- 2301 Drum Storage
- 2303 Drum Storage

- 2870 Multiplexer Channel with #6990, #6991, #6992
1, 2, 3 Selector Subchannels

Devices Used Only by an Operating System in a
Virtual Machine and not by CP-67

- 2321 Data Cell Drive

- 2400 Magnetic Tape Units

- 2250 Display Unit
- 2260 Display Station

- 2860 Selector Channel with
#1850 Channel-to-Channel Adapter

- 2780 Data Transmission Terminal
- 1130 Computing System

APPENDIX E

Devices Supported by CMS

Core size: Minimum 256K or up in multiples of 8K (up to 16M virtual)

1052 Printer-Keyboard

Six 2311 Disk Storage Drives or

2314 Direct Access Storage Facility

(2 disk storage modules minimum)

2540 Card Reader/Punch

1403 Printer

Two 2400 series tape drives, nine or seven track

200, 556, 800 or 1600 bpi

(one nine track, 800 or 1600 bpi required for installation)

APPENDIX F

CP-67 Restrictions

The virtual machine created by CP-67 is capable of running an operating system as long as that system does not include any timing dependencies or dynamically modified channel programs. Dynamically modified channel programs are those which are changed between the time the SIO is issued and the end of the I/O operation; i.e., changed by the channel program or the CPU. However, the OS/360 Indexed Sequential Access Method (ISAM), which uses a self-modifying channel program for some of its operations, receives special handling if CP-67 is generated with the ISAM option, and the virtual machine using ISAM has that option selected in the directory description.

In addition to the above restriction of dynamically modified I/O sequences, there are the following restrictions for non-dedicated direct access devices; i.e., those described in the directory.

- a. In the case of a channel program for which the SIO command results in a condition code of 1 (Channel Status Word stored) with the busy bit off, the same operation on a non-dedicated direct access device will result in a condition code of 0 and subsequent interrupt. This is because CP-67 prefixes a SEEK command to all channel programs for non-dedicated direct access devices to re-position the arm to the track last accessed by the user.
- b. In the case of Read Home Address with the skip bit off, CP-67 will modify the home address data in user core at the completion of the channel program because of CP's having to relocate addresses for mini-disks; therefore, the data buffer area may not be dynamically modified during the I/O. If Read Home Address is issued with the skip bit on, CP will not modify the home address data but the direct access device will be positioned to the appropriate record.
- c. On a mini-disk, if a CCW string uses multi-track search on I/O operations, subsequent operations to that disk must have preceding seeks or continue to use multi-track operations. There is no restriction for dedicated disks.

Other restrictions that exist in CP-67 are as follows:

1. If a CCW that is issued by the virtual machine with the CC and SILI bits on references data which crosses a page

- boundary, it will be expanded into multiple CCWs by CP. If the residual count is greater than or equal to the count in the last generated CCW, command chaining will be suppressed.
2. Data chaining on the 2301 Drum Storage will cause overrun problems, as this is a hardware restriction. Hence, if user files are placed on the 2301, any I/O that crosses a page boundary will produce chaining checks. There may also be a potential problem with the 2303 Drum Storage Unit.
 3. If one CCW issues I/O for less than eight bytes, a chaining check may occur for some devices, in particular the 2314. This is also a hardware restriction. This problem can arise when CP-67 translates and expands CCWs to data chain into or out of non-contiguous pages. One solution is to create I/O buffers in multiples of a double word and place them on a double word boundary.
 4. CP-67 has no path finding and hence will not take advantage of the two channel switch. However, a two channel switch can be used between the Model 67 running CP-67 and another CPU.
 5. If a machine check occurs while the virtual machine is running, the logout area is not returned to the virtual machine; therefore, any error recovery procedures that might exist in the virtual machine may not work properly once the virtual machine continues running.
 6. The DIAGNOSE command cannot be issued by the virtual machine for its normal function. CP-67 uses the DIAGNOSE command to allow the virtual machine to communicate with it such as in virtual console functions.
 7. Dynamic buffering, such as in the Queued Telecommunications Access Method (QTAM), Basic Telecommunications Access Method (BTAM), and Telecommunications Access Method (TCAM), violates the restrictions on timing dependence and dynamically modified channel programs and are not supported.
 8. The PCI plus channel end interrupt will always be presented simultaneously to the virtual machine whereas this may not be true on the real machine.
 9. Program Controlled Interrupt (PCI) is timing dependent. The PCI interrupt will be returned to the virtual machine but always upon completion of the channel program, thus OS/360 PCI Fetch and chain scheduling will not provide the performance improvements obtained on a real machine.
 10. Because of timing dependencies, CP-67 will not give a control-unit-busy condition on a SIO to a virtual machine,

even though that may be expected.

11. CP locks a page in core whenever I/O is being performed on that page. The lock count cannot be greater than 65,535. I/O with more than 65,535 CCWs specifying data addresses into the same page cannot be run.
12. The number of pages into which I/O is being performed must not exceed the total number of user pages available in real memory.
13. Halt I/O to a non-busy channel does not reach the channel, as in the channel-to-channel adapter used by ASP.
14. The CP console function IPL is not a true IPL; it is a simulator of most IPL sequences BUT not all. To determine which IPL sequences CP will not simulate properly, see the listing of the IPL module. The IPL simulator loads into either halfway up the virtual machine aligned on a page boundary or 20000 hex, whichever is less; therefore, the IPL simulator could possibly be overlaid by some IPL sequences.
15. Emulators are not supported by CP-67.
16. The interval timer provided by CP-67 does not perform in the same manner as the standard interval timer. The 'real timer' that is given to a virtual machine by specifying the real timer option in the directory reflects the execution time plus the wait time for that virtual machine. If a virtual machine does not have the real timer option, its timer reflects only the execution time used. A system such as APL needs the real timer, as it is timer driven.
17. Punch-feed-read is not supported.
18. For spooled I/O devices, the following are not supported: stacker selection and detection of channels in the printer's carriage control tape.
19. Paper tape I/O is not supported by CP from the virtual machine operator's console.
20. The 1051/1052 Model 2 Data Communication System is supported only as a keyboard operator's console. Paper tape I/O and other modes of operation are not recognized and hence may not work properly.
21. CP does not support count control on the virtual 1052 operator's console.
22. The pseudo-timer device OFF of type TIMR does not return an

| interrupt from the SIO; therefore do not use EXCP to read this device.

23. The virtual machine which is created for running CP-67 has a simplex CPU with 24 bit addressing; it is a replica of a Model 67.

CP-67/CMS System Description Manual

INDEX

Abbreviations.....	16, 27
Access methods.....	
Accounting.....	21
APL.....	17, 36
ASP.....	17, 36
Assembler.....	14, 16, 22
ATTN.....	8
Batch monitor.....	16, 4
BRUIN.....	14, 22
Chaining check.....	35
Commands.....	2, 14, 21
Compiler(s).....	14, 16
Configuration(s).....	29, 1, 4, 6, 10
Console functions.....	2, 8, 18
Context editor.....	14
Conversational.....	1, 4, 9, 16
Core requirements.....	9, 16
CPS.....	17
Debugging.....	15, 4, 14
Desk calculator.....	14, 22
DIAGNOSE.....	2, 35
Diagnostics.....	2, 14, 17
DIAL.....	2, 8, 17, 19
Directory.....	6, 11, 24, 25
Disk(s).....	3, 6, 10, 14, 17, 25
Disk configuration.....	29, 33
DOS.....	17
Dynamic address translation.....	7
Emulators.....	36
File identification.....	11
File management.....	10
FORTRAN.....	24, 14
HASP.....	17
Interrupt.....	7, 15, 30, 35
I/O Error handling.....	9
IPL.....	8, 19, 2, 11, 36
Language processors.....	14
Libraries.....	16, 11, 14
LOGIN.....	25, 19, 26
LOGOUT.....	19, 25
Mini-disk.....	3, 17
Nucleus.....	11, 16, 2, 26
Operator.....	1, 4, 8, 17, 18
Operator's console.....	2, 8, 9
Operating system(s).....	1, 4, 8, 17, 18
OS.....	18, 14, 17
Paging.....	6
Password.....	11
Performance.....	18
PL/I.....	26

Privilege class.....6
Problem state.....7
Pseudo-timer device.....36, 36
Restrictions.....34
SNOBOL.....14, 27
Spooling.....8, 18, 20
Supervisor state.....7
SYSERR.....9
Terminal types.....8, 30, 31
Terminal use.....2, 1, 6, 10, 17
Text processor.....14
Time slice.....36, 36
User profile.....25
Userid(s).....6, 11
Utilities.....14
Virtual CPU.....2
Virtual I/O.....2, 8, 8
Virtual machine.....2
Virtual memory.....2
Warm start.....8

IBM

**International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)**

**IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)**

READER'S COMMENT FORM

Control Program-67/Cambridge Monitor System
(CP-67/CMS) Version 3.1
System Description Manual

GH20-0802-2

Please comment on the usefulness and readability of this publication, suggest additions and deletions, and list specific errors and omissions (give page numbers). All comments and suggestions become the property of IBM. If you wish a reply, be sure to include your name and address.

COMMENTS

—
fold

—
fold

—
fold

—
fold

• Thank you for your cooperation. No postage necessary if mailed in the U.S.A.
FOLD ON TWO LINES, STAPLE AND MAIL.

YOUR COMMENTS PLEASE...

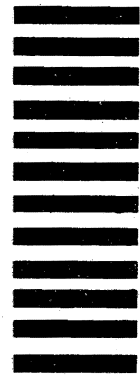
Your comments on the other side of this form will help us improve future editions of this publication. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material.

fold

fold

FIRST CLASS
PERMIT NO. 1359
WHITE PLAINS, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES



POSTAGE WILL BE PAID BY...

IBM Corporation
1133 Westchester Avenue
White Plains, N.Y. 10604

Attention: Technical Publications

fold

fold



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
[U.S.A. only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

Control Prog: 67/Cambridge Monitor System (CP-67/CMS) V. 3.1 SDM Printed in U.S.A. GH20-0802-2