File No. S370-20
Order No. GC20-1800-4

Systems

# IBM Virtual Machine Facility/370: Introduction

Release 2 PLC 13

This publication introduces VM/370, and defines the minimum equipment configuration necessary to execute it. It is intended for anyone who is interested in VM/370. However, the reader should have a basic understanding of IBM data processing.

VM/370 (Virtual Machine Facility/370) is an operating system that manages the resources of a single System/370 computer so that multiple computing systems (virtual machines) appear to exist. VM/370 consists of a Control Program (CP), which manages the real computer, a Conversational Monitor System (CMS), which is a general-purpose conversational time-sharing system that executes in a virtual machine, and a Remote Spooling Communications Subsystem (RSCS), which spools files to and from geographically remote locations.

The first section of the publication is an introduction; it describes what VM/370 can do. The second, third, and fourth sections describe the Control Program, Conversational Monitor System, and Remote Spooling Communications Subsystem, respectively. The appendixes include information about Recovery Management Support, system requirements, supported language processors and emulators, compatibility of VM/370 with CP-67/CMS, and VM/370-related publications for CMS users.

This publication is a prerequisite for the VM/370 system library.

IBM

*Fifth Edition* (January 1975)

This is a major revision of GC20-1800-2 and makes that edition and Technical Newsletter GN20-2638, as well as GC20-1800-3, obsolete. This edition, together with Technical Newsletter GN20-2657, dated March 31, 1975, corresponds to *Release 2 PLC 13* (Program Level Change) of IBM Virtual Machine Facility/370 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters.

Changes are periodically made to the specifications herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/360 and System/370 Bibliography*, Order No. GA22-6822, and its *Virtual Storage Supplement*, Order No. GC20-0001, for the editions that are applicable and current.

Technical changes and additions to text and illustrations are indicated by a vertical bar to the left of the change.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, VM/370 Publications, 24 New England Executive Park, Burlington, Massachusetts 01803. Comments become the property of IBM.

# Preface

This publication introduces and describes the IBM Virtual Machine Facility/370 (VM/370) and its components, the Conversational Monitor System (CMS) and the Remote Spooling Communications Subsystem (RSCS).

This publication contains four parts:

- The "Introduction" describes VM/370, virtual machines, and their applications.
- "Control Program Description" describes how the VM/370 control program manages the resources of the real computing system.
- "Conversational Monitor System" describes the facilities of CMS: problem solving and program development capabilities for interactive users.
- "Remote Spooling Communications Subsystem" describes the functions and organization of RSCS.

This publication contains five appendixes:

- Appendix A: Recovery Management Support
- Appendix B: System Requirements
- Appendix C: Language Processors and Emulators
- Appendix E: Compatibility of VM/370 with CP-67/CMS
- Appendix F: VM/370-Related Publications for CMS Users

The reader must have a basic knowledge of data processing systems and an understanding of virtual storage concepts. See the student text publication *Introduction to Virtual Storage in System/370*, Order No. GR20-4260, for information about virtual storage.

When the term 3330 is used in this publication, it refers to the IBM 3330 Disk Storage, Models 1, 2, and 11, or the IBM 3333 Disk Storage and Control, Models 1 and 11.

References to the IBM 2741 Communication Terminal also include the IBM 3767 Communication Terminal (in 2741 mode), unless otherwise specified.

RELATED PUBLICATIONS

*IBM Virtual Machine Facility/370*:

*Planning and System Generation Guide*, Order No. GC20-1801

*Command Language Guide for General Users*, Order No. GC20-1804

*EDIT Guide*, Order No. GC20-1805

*Operator's Guide*, Order No. GC20-1806

*System Programmer's Guide*, Order No. GC20-1807

*System Messages*, Order No. GC20-1808

*OLTSEP and Error Recording Guide*, Order No. GC20-1809

*Terminal User's Guide*, Order No. GC20-1810

*EXEC User's Guide*, Order No. GC20-1812

*Glossary and Master Index*, Order No. GC20-1813

*Release 2 Guide*, Order No. GC20-1815

*Remote Spooling Communications Subsystem (RSCS) User's Guide*, Order No. GC20-1816.

*Control Program (CP) Program Logic*, Order No. SY20-0880

*Conversational Monitor System (CMS) Program Logic*, Order No. SY20-0881

*Service Routines Program Logic*, Order No. SY20-0882

*Remote Spooling Communications Subsystem (RSCS) Program Logic*, Order No. SY20-0883.

*Quick Guide for Users*, Order No. GX20-1926

*Summary of CP and CMS Commands*, Order No. GX20-1961

*Note*: The *Quick Guide* and *Summary of CP and CMS Commands* are part of Order No. GBOF3576.

Figure 1 is an overview of the VM/370 library, with the publications grouped according to their probable users.

References in the text to titles of related VM/370 publications will be given in abbreviated form.

## SUPPLEMENTARY PUBLICATIONS

Titles of other publications for VM/370 users are in "Appendix F: VM/370-Related Publications for CMS Users."

**Virtual Machine Facility/370 (VM/370) Library**

*(Release 2 PLC 11)*

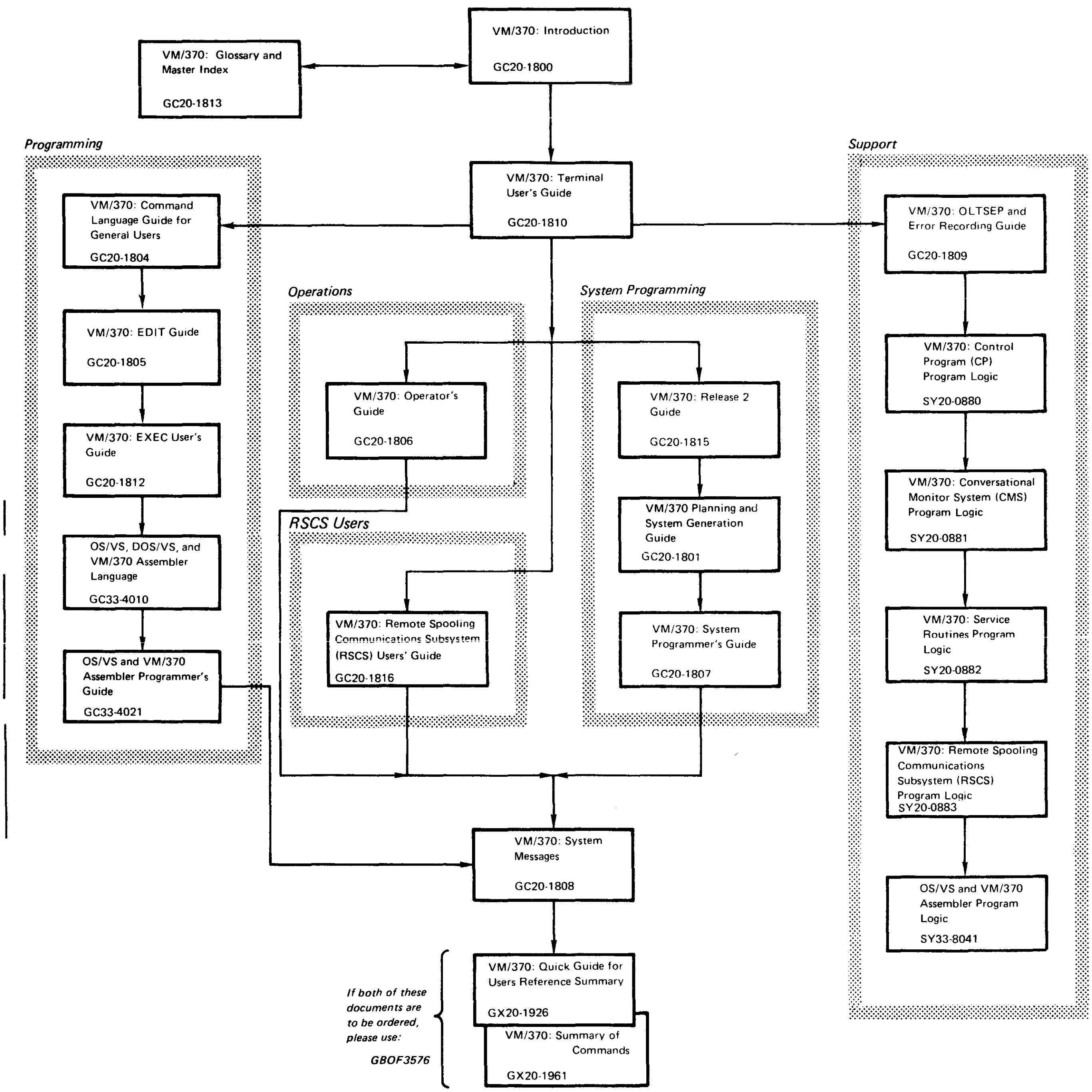


**Figure 1. Virtual Machine Facility/370 Library**

Summary of Amendments
for GC20-1800-4
as updated by GN20-2657
VM/370 Release 2 PLC 13

## REMOTE 3270 SUPPORT

New: Program Feature

VM/370 now supports the 3270 Display System as a remote virtual machine console attached via nonswitched point-to-point communications lines to a 2701 Data Adapter Unit or a 2703 Transmission Control Unit. The 3704/3705 Emulation Program (EP) and the Partitioned Emulation Program (PEP), in emulation mode, also support the remote 3270.

This program feature also allows the remote 3270 user to copy an entire screen display on a 3284, 3286, or 3288 printer at the remote location.

"Appendix B. System Requirements" is updated to reflect this support.

## VM/370 MEASUREMENT FACILITY

New: Program Feature

The VM/370 control program has two commands that measure system performance. The MONITOR command is enhanced to collect system performance data over a period of time and to sample system loading and activity for system performance analysis. A new command, INDICATE, displays system loading conditions.

A new section, "Performance Measurement and Analysis" is added to the "Control Program Description" portion of this manual.

## VM/VS HANDSHAKING

New: Program Feature

The VM/VS Handshaking feature is a communication path between VM/370 and OS/VS1 that makes each system control program aware of certain capabilities and requirements of the other.

A new section, "VM/VS Handshaking", is added to the "Control Program Description" portion of this manual.

## MISCELLANEOUS

Changed: Documentation

"Appendix D: VM/370 Restrictions" is removed. It is now in the *VM/370: Planning and System Generation Guide*.

## NEW DEVICE SUPPORT

New: Program Feature

VM/370 now supports the IBM 3340 Direct Access Storage Facility. This support includes the 35-megabyte and 70-megabyte 3348 Data Modules, Rotational Position Sensing (RPS), and the Fixed Head Feature.

VM/370 now supports the IBM 3767 Communication Terminal (at 300 bps), operating as an IBM 2741 Communication Terminal only.

## REMOTE SPOOLING COMMUNICATIONS SUBSYSTEM (RSCS)

New: Program Feature

The Remote Spooling Communications Subsystem (RSCS) is now supported as a VM/370 component. RSCS can transmit spool files across a teleprocessing network to and from remote stations. Remote stations supported include HASP-type and ASP-type batch processors and work stations. The new component is a control program designed to run in a virtual machine dedicated to remote spooling.

The subsystem can be controlled by virtual machine users, the RSCS virtual machine operator, and operators at remote stations.

## OS ACCESS ENHANCEMENT FOR READING DOS FILES

New: Program Feature

CMS can now read, but not write or update, DOS sequential files that reside on DOS disks. The OS macros simulated by CMS are used to read DOS data.

## SELECTION OF VIRTUAL MACHINE CHANNEL OPERATING MODE

New: Program Feature

Programs using block multiplexer channel operation (such as DOS/VS, VS1, and VS2) can now be run in block multiplexer mode. The virtual machine's channel operating mode can now be selected by the user of the virtual machine or it can be set by the Directory service program. The mode selected can be either selector or block multiplexer.

## PUBLICATION CHANGES

Changed: Documentation Only

The title "Appendix C. IBM Programs Executable under CMS" has been changed to "Appendix C. Language Processors and Emulators."

"Appendix D. System Commands" has been deleted; the information in it has been moved to the sections "Control Program Description" and "Conversational Monitor System."

The title "Appendix F. VM/370 Compatibility" has been changed to "Appendix E. Compatibility of VM/370 with CP-67/CMS."

The Glossary has been deleted. A comprehensive list of data processing terms appears in the *IBM Data Processing Glossary*, Order No. GC20-1699. A comprehensive list of VM/370 terms appears in the *VM/370: Glossary and Master Index*.

IBM 3704/3705 COMMUNICATIONS CONTROLLERS NETWORK CONTROL PROGRAM (NCP) AND PARTITIONED EMULATION PROGRAM (PEP)

New: Program Feature

VM/370 now supports all three of the 3704/3705 control programs:

- Emulation Program (EP)
- Network Control Program (NCP)
- Partitioned Emulation Program (PEP)

The Preface is updated to remove a statement limiting support to the EP control program.

## NEW DEVICE SUPPORT

New: Program Feature

The following devices are now supported:

- IBM 3066 System Console, Model 2
- IBM 3272 Control Unit, Model 2 (local attachment)
- IBM 3277 Display Station, Model 2 (local attachment)
- IBM 3330 Disk Storage, Model 11
- IBM 3333 Disk Storage and Control, Model 11
- IBM 3420 Magnetic Tape Unit, Models 4, 6, and 8 (6250 bpi density)

## READ-ONLY OS DATA SET ACCESS SUPPORT

New: Program Feature

CMS can now read, but not write or update, real OS sequential and partitioned data sets.

## IBM 3704/3705 COMMUNICATIONS CONTROLLERS

New: Program Feature

This publication now includes information about use of the IBM 3704 and 3705 Communications Controllers in Network Control Program mode and Partitioned Emulation Program mode as well as in Emulation Program mode.

## VIRTUAL MACHINE ASSIST FEATURE

New: Program Feature

The Virtual Machine Assist feature, available on the System/370 Models 135, 145, and 158, reduces VM/370 overhead associated with running OS/VS1 and DOS/VS operating systems in virtual machines. This is described under "Virtual Machine Assist Feature."

## SVC 76 ERROR RECORDING INTERFACE

New: Program Feature

By use of the SVC 76 Error Recording Interface, VM/370 provides uniform recording of errors encountered by some operating systems running in virtual machines. This is described in "Appendix A: Recovery Management Support."

## ENHANCEMENTS TO THE CMS EDITOR

New: Program Feature

The section entitled "Manipulation of User Files" has been changed to reflect enhancements to the CMS Editor associated with support of display devices.

## NEW PROGRAM PRODUCTS

New: Programs

The OS PL/I Checkout Compiler, the VS BASIC Processor, and the Planning Systems Generator have been added to the list of Program Products executable under CMS.

# Contents

FIGURES

# Introduction

Virtual Machine Facility/370 is a system control program (SCP) that manages a real computing system so that all its resources -- CPU, storage, and input/output devices -- are available to many users at the same time. Each user has at his disposal the functional equivalent of a real, dedicated computing system. Because this functional equivalent is simulated for the user by VM/370 and does not really exist, it is called a "virtual" machine.

VM/370 is designed for IBM System/370 Models 135, 145, 155 II, 158, 165 II, and 168. The real System/370 must have the Dynamic Address Translation feature, a hardware facility that translates virtual storage addresses to real storage addresses, and the System Timing Facility. Also, it must operate in extended control mode, a mode in which all the features of a System/370, including dynamic address translation, are operational.

VM/370 is the System/370 version of a control program called CP-67/CMS, which performs similar functions on a System/360 Model 67. Like its predecessor, VM/370 provides:

1. Virtual machines and virtual storage.
2. The ability to run multiple operating systems concurrently.
3. A conversational, time-sharing system.

A major difference between CP-67/CMS and VM/370 is that VM/370 provides a Remote Spooling Communications Subsystem (RSCS). In addition, VM/370 supports such devices as the IBM 3330 Disk Storage, the IBM 3340 Direct Access Storage Facility, and the IBM 2305 Fixed Head Storage, and offers several performance options to optimize performance in the virtual machine environment.

## VM/370 Components: CP, CMS, and RSCS

VM/370 has three major elements:

1. The Control Program (CP), which controls the resources of the real computer to provide multiple virtual machines.
2. The Conversational Monitor System (CMS), a subsystem that gives users a wide range of conversational, time-sharing facilities, including creation and management of files, and compilation, testing, and execution of problem programs.
3. The Remote Spooling Communications Subsystem (RSCS), which enables users to transmit files to and receive files from remote stations in the RSCS teleprocessing network.

## Virtual Machine Operating Systems

While the control program of VM/370 manages the concurrent execution of the virtual machines, it is also necessary to have an operating system managing the work flow within each virtual machine. Because each virtual machine executes independently of other virtual machines, each one may use a different operating system, or different releases of the same operating system.

The operating systems that can run in virtual machines are shown in Figure 2. CP provides each of these with virtual device support and virtual storage. The operating systems themselves execute as though they were controlling real devices and real storage, but they must not violate any of the restrictions listed in the VM/370: Planning and System Generation Guide.

| Batch or Single-User Interactive | Multiple-Access |
|---|---|
| DOS | APL\DOS-360 |
| DOS/VS | (with CP option) |
| OS/PCP | VM/370 |
| OS/MFT | Time Sharing |
| OS/MVT | Option of OS |
| OS/VS1 | |
| OS/VS2 | Conversational |
| OS-ASP | CMS |
| PS44 | |
| RSCS | |

Figure 2. Virtual Machine Operating Systems

Figure 3 shows six virtual machines running concurrently under control of CP on an IBM System/370, Model 145, with 512K of real storage. One machine is doing batch production work under the present release of DOS; a second is executing programs that require a back release of DOS; and a third is controlling the RSCS network. The other three virtual machines are running CMS for three separate conversational users.

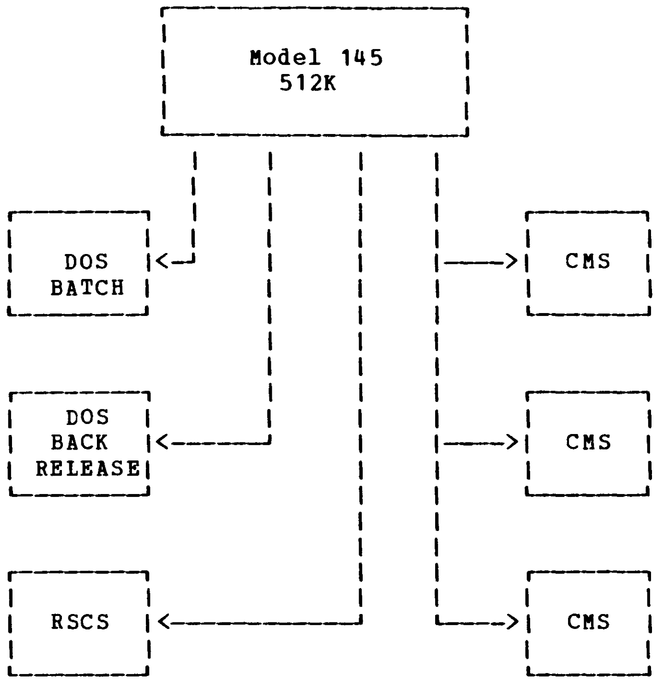


Figure 3. Multiple Virtual Machines

## The VM/370 Directory

The VM/370 directory is a disk-resident file that contains entries for all the virtual machines in the VM/370 system. Each directory entry contains a user identification (userid), a password, the storage size of the virtual machine, the privilege class or classes assigned to the user, and the I/O devices he can use. It may also contain other optional information. When the user logs on the VM/370 system by entering a valid userid and password, a virtual machine is created for him based on the information in his directory entry. He then can immediately begin to use his virtual machine.

Figure 4 shows a typical logon procedure of a user whose userid is "smith." When he types in his password, printing or displaying of the password may be masked, to maintain security. CP accepts his userid and password as correct, and notifies him that he is logged on. The user then loads an operating system, in this case, CMS.

```
vm/370 online
logon smith
ENTER PASSWORD:

LOGON AT 11:03:18 ON WEDNESDAY 01/15/75

ipl cms
```

Figure 4. Logging onto VM/370 and Invoking CMS

## Virtual Machine Components

The components of a virtual machine configuration are:

- Virtual System Console
- Virtual Storage
- Virtual CPU
- Virtual Channels and I/O Devices

Each user's entry in the VM/370 directory defines the devices and the amount of storage he needs. Descriptions of the components follow.

### VIRTUAL SYSTEM CONSOLE

The user's terminal, such as the IBM 2741 Communication Terminal or the IBM 3277 Display Station, serves as the virtual system console. Using the terminal keyboard and CP commands, a user can perform almost all the functions an operator can perform on a real machine. These include loading an operating system, stopping and starting virtual machine execution, and displaying and changing the contents of registers and storage.

### VIRTUAL STORAGE

Each virtual machine has its own virtual storage space, which may be as small as 8K (8192) bytes or as large as 16 million bytes, or any size in between that is a multiple of 4K (4096) bytes. It uses this storage space exactly as though it is real storage, but it is not aware of or limited by the storage size of the real machine. For example, three virtual machines of 256K bytes each can run on a single real computing system that has only 384K bytes of real storage. This is possible because CP brings into real storage whatever part of virtual storage is needed for the virtual machine's execution, while parts that are not needed immediately may not be kept in real storage. Instead, they may be sent to a direct access device and stored until they are needed again.

The size of the virtual storage is defined in the virtual machine's directory entry and may differ among virtual machines.

Each virtual machine can refer only to its own virtual storage. This protects each virtual machine's storage from the activities of other virtual machines.

VIRTUAL CPU

CP provides CPU resources to each active virtual machine through time slicing. Each virtual CPU periodically gets a share of real CPU time.

Essentially, the virtual CPU provides the facilities described in _IBM System/370 Principles of Operation_, Order No. GA22-7000. Some restrictions exist, and are discussed in the _VM/370: Planning and System Generation Guide_.

The virtual machine operating system can be either single task or multitask. The virtual CPU can be run in either basic or extended control mode (extended control mode includes all the facilities necessary to run VM/370 as the virtual machine's operating system). Thus OS/MFT and OS/VS1, as well as CMS and VM/370, can all run in virtual machines.

The virtual machine can execute all System/370 instructions except READ DIRECT and WRITE DIRECT. The DIAGNOSE instruction is reserved for special program communication with CP.

VIRTUAL CHANNELS AND I/O DEVICES

A virtual machine can support the same devices as a real machine can. Virtual devices are logically controlled by the virtual machine and not by VM/370. Input/output (I/O) operations, and any error recovery processing, are normally the complete responsibility of the virtual machine operating system.

Virtual and real device addresses may differ. CP converts virtual channel and device addresses to real channel and device equivalents and performs whatever data translations are necessary.

All virtual devices must have real counterparts, such as disk to disk and tape to tape. Some virtual devices (like tapes) must have a one-to-one relationship with a real counterpart. Others may be mapped to portions of a real device (for example, a virtual disk may occupy all or part of a real disk). Several virtual disks can exist on one real disk.

Figure 5 illustrates a real 2319 disk volume that has been mapped as three virtual disks. These disks may belong to three virtual machines or to one virtual machine. Such virtual disks are called minidisks because they are not mapped to full-sized real equivalents. Service programs distributed with VM/370 create and maintain these minidisks, which can be used as normal disk volumes by CMS, DOS, and OS.

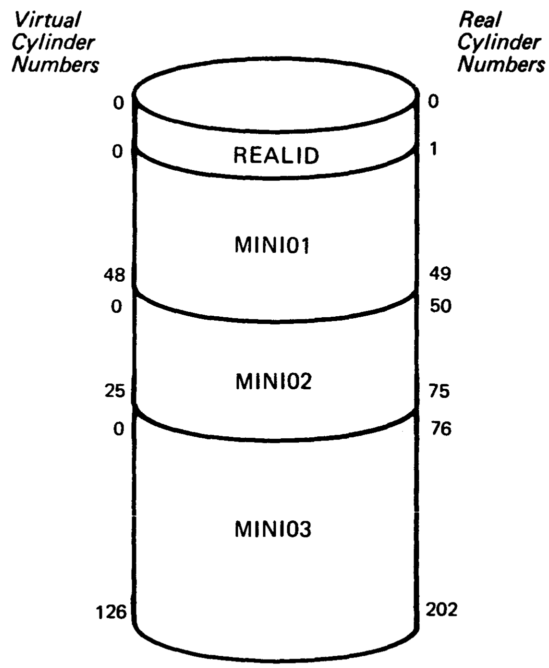


Figure 5. Real Disk Partitioned into Minidisks

REALID is the real volume serial number of this volume. MINI01, MINI02, and MINI03 are the labels of minidisks on the volume REALID. Note that each minidisk starts at virtual cylinder zero.

A virtual machine configuration may include virtual unit record devices, such as printers and punches. Since virtual machines usually have low volume requirements for unit record I/O, the input and output of several virtual unit record devices can usually be handled by one real unit record device. CP controls these virtual devices (as well as the real devices) and directs all input and output for them to intermediate direct access space that has been allocated for this purpose. This function is called spooling and is explained under "Spooling" in the section entitled "The Control Program."

A virtual device that is defined as dedicated to a specific virtual machine must have a real equivalent. The virtual machine, not CP, then controls both the real and virtual device.

A virtual machine configuration can include a virtual transmission control unit (TCU). A subset of the lines of a real TCU may be defined as a virtual TCU, as shown in Figure 6.

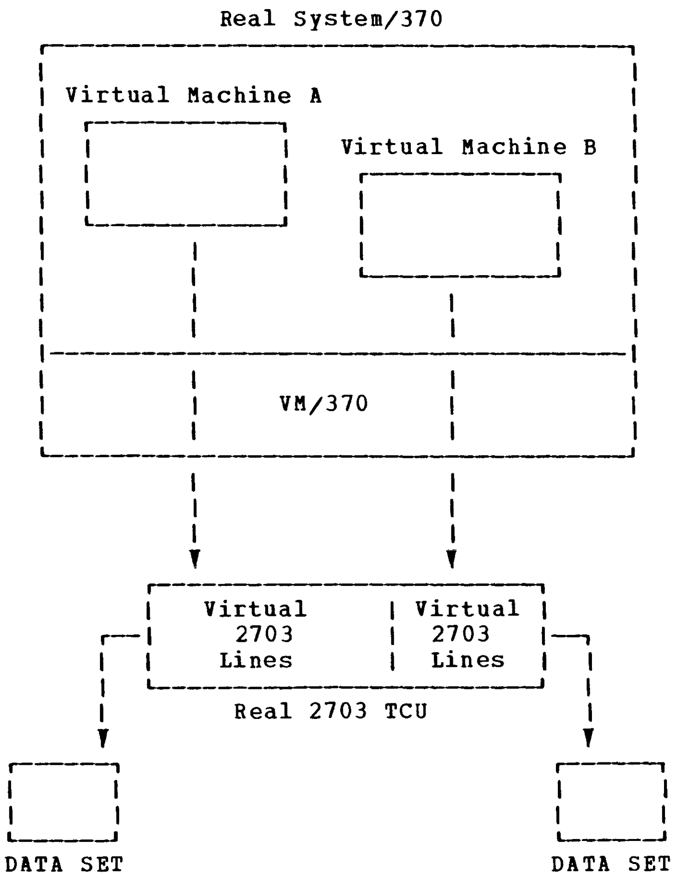


Figure 6. Virtual Devices: Transmission Control Units

A virtual channel-to-channel adapter can be defined either with or without a real equivalent; the former permits a virtual machine to communicate with a real computing system other than its own, while the latter allows direct communication between virtual machines in the same computing system.

Restrictions that apply to virtual machines are stated in the *VM/370: Planning and System Generation Guide*.

Factors that influence the performance of virtual machines, and options that can be used to improve performance, are described in the section entitled "Performance."

## VM/370 Applications

VM/370 assists an installation to perform its work more efficiently and easily. Virtual machine applications aid programmers, operations personnel, and interactive users.

PROGRAMMING

Programming is facilitated in the following ways:

1. Programs being developed need not conform to the real storage size of the real computer.
2. Virtual machines make program testing more flexible. Subject to available resources, a virtual machine can be made active whenever needed, thus relaxing tight testing schedules and allowing programmers more compilations and tests per day.
3. JCL (Job Control Language) usually is not needed when compiling, assembling, and/or testing under CMS.
4. Users can test privileged code in their own virtual machines.
5. Programmers can use debugging aids at their terminal that parallel those of an operator at a system console: displaying and storing into the general or floating-point registers or into virtual storage, instruction address stopping, and altering the normal flow of execution. Which of these functions each user is allowed to perform are defined by the privilege class(es) assigned to him. (For a discussion of privilege classes, see "CP Commands.")
6. CMS simplifies the creation and manipulation of source programs on disk, and allows the user to examine selected portions of program listings and storage dumps at his terminal.

7. RSCS enables users to transmit files to and receive files from users at other geographic locations.

8. The VM/370 data privacy, security, and user-isolation features protect each user's data, programs, and disk files from access or destruction by other users.

9. Many System/360 and System/370 programs can be compiled under control of CMS; within certain restrictions these programs may also be tested under CMS. (Refer to "Program Development and Execution" in the section "Conversational Monitor System" for a more complete discussion of program execution under CMS.) DOS assembler language programs can be compiled under CMS if the installation adds the appropriate DOS macros to the CMS system. Problem programs using DOS macros can be conversationally developed under control of CMS; then control of the virtual machine is passed to DOS, and the programs are compiled and tested. The user specifies which operating system is to control his virtual machine by means of the IPL command of CP.

## OPERATIONS

The virtual machine environment relieves certain problems of scheduling, support, and backup, and expedites production in the following ways:

1. System generation, support, and testing, as well as operating system conversion and testing, can be done without a dedicated real machine, concurrently with normal production work. This reduces errors that might otherwise be caused by using a system that has not been fully tested, and it also reduces the possibility of abnormal terminations of the system. For example, a program temporary fix (PTF) applied to a modifiable copy of an IBM operating system volume can be tested concurrently with the production execution of that same operating system in another virtual machine, provided sufficient direct access storage resources are available. The virtual machine test will be analogous to one made on a real machine providing:

   - There are no timing dependencies.
   - The test is not measuring time.
   - Dynamically modified channel programs are not used except as noted in "Appendix D: VM/370 Restrictions."

   A possible combination of virtual machines in a VM/370 configuration is shown in Figure 7. Operating system testing is shown running concurrently with batch work and a variety of conversational applications.

2. VM/370 allows DOS and OS, including virtual storage (VS) versions, to run concurrently on the same System/370. Multiple copies of the same operating system can also run concurrently in separate virtual machines.

3. Many types of batch applications can be run, either in an individual user's virtual machine or in a virtual machine dedicated to running batch, with no change to the batch program.

4. New computer operators can get "hands on" experience using a virtual machine terminal as a system console.

5. An installation using VM/370 has more flexibility in using another System/370 computing system for backup. The backup system need not be the same System/370 model nor have the same amount of real storage. Backup may be done in two ways:

   - The VM/370 system residence volume and the user and CMS volumes may be run on another System/370 if the device addresses on both machines are the same. This is not unique to VM/370; the same procedure would be used to back up OS or DOS systems.
   - The second method is unique to VM/370. The user volumes alone may be carried over to another computing system that is using VM/370. The backup system must include, but is not limited to, the same type and number of real devices as these user volumes require. Since the virtual devices defined in the user volumes are not assigned to specific real devices until execution time, the installations need not concern themselves with device addresses; VM/370 on the backup system will assign real devices just as it does for its own virtual machines. Thus, the production work of the system being backed up can be run in virtual machines concurrently with the execution of the virtual machines of the backup system.

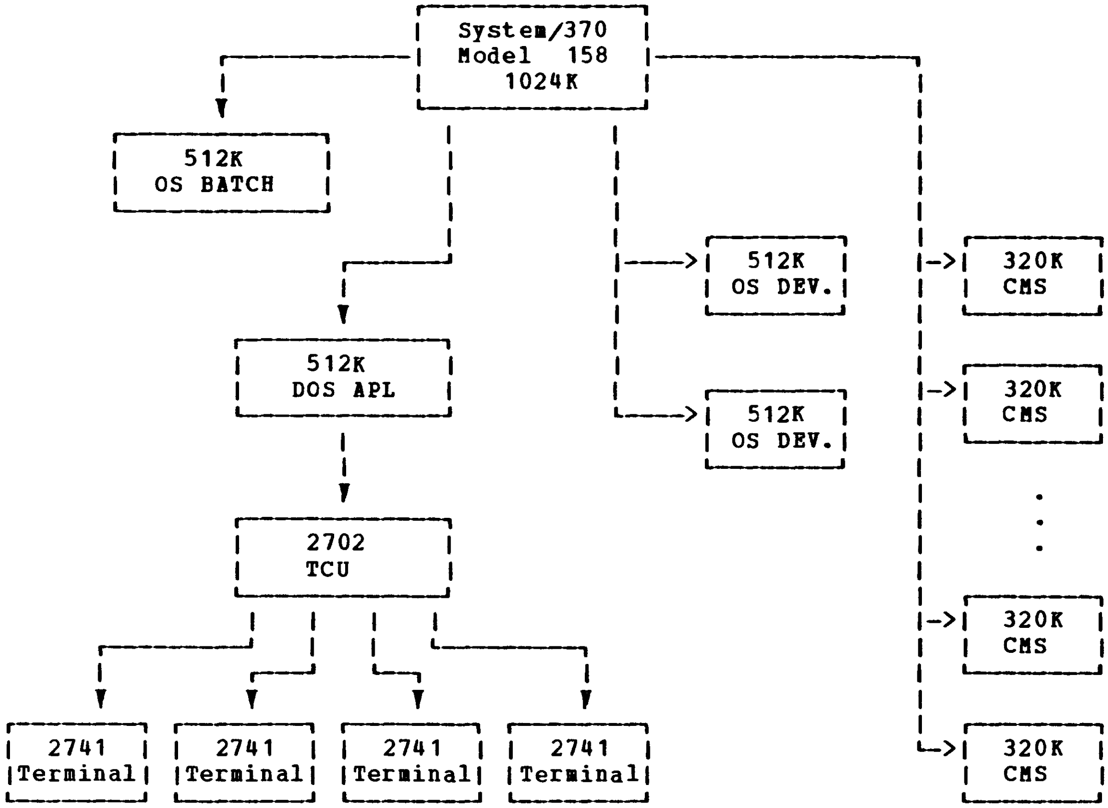


Figure 7. Virtual Machines for Concurrent Production, Development, and Testing

## INTERACTIVE USE

There are two kinds of interactive systems that run under VM/370: multiple-access and single user.

1. Multiple-access systems like APL\DOS-360 run in one virtual machine and directly service many interactive terminals. A user of a multiple-access system issues the DIAL command instead of the LOGON command to connect his terminal with the virtual machine running the multiple-access system he wishes to use. Once his terminal is connected, the user issues statements in the command language associated with the multiple-access system only.

   For example, the command DIAL APL could connect the user's terminal with an APL\DOS-360 system running in a virtual machine under VM/370. Once connected, the user would communicate only with APL commands.

   Note: The user should be aware that when he uses the DIAL command, he does not log on to CP and therefore he cannot use any CP commands.

2. Systems that can be run interactively by a single user include the Conversational Monitor System and any operating system that can run on a virtual machine. A time sharing environment is created when VM/370 creates multiple virtual machines, each controlled by an operating system, for example, CMS. These systems operate concurrently with each other as well as with other conversational or batch systems. CMS is useful for program development and problem solving.

The CMS command language provides each user with a wide range of capabilities at his terminal, such as:

- Creating source programs, data, and text files directly on disk.
- Adding, deleting, modifying, rearranging, extracting, or merging files and/or portions of files.

- Compiling, testing, and debugging some types of OS problem programs under CMS.
- Creating complete job streams to be passed to batch operating systems such as DOS or OS for compilation and/or execution. The resultant output can be printed on a high-speed printer or directed back to CMS for analysis and correction by the user.
- Submitting jobs to a background CMS Batch facility.
- Extending CMS facilities to suit the user's requirements, such as creating additional commands or developing command procedures.

# Control Program Description

The control program of VM/370, CP, creates and controls virtual machines. A virtual machine is the functional equivalent of a real computing system. Executing a program on a virtual machine produces exactly the same output as executing that program on a real machine.

When a user logs onto VM/370, CP creates a virtual machine for him based on information stored in the VM/370 directory. The VM/370 directory contains one entry for each user identification. Each entry includes: the password associated with the userid; a description of the virtual input/output devices associated with this virtual machine; its normal and maximum virtual storage sizes; the user's CP command privilege class(es); and optional virtual machine characteristics, such as extended control mode.

CP controls the resources of the real computer to provide multiple virtual machines. CP intercepts, translates, and schedules all real input/output operations of the virtual machine. All virtual machines execute in problem state, and the control program traps and processes all interrupts and privileged instructions. Only CP executes in supervisor state.

## Virtual Machine Time Management

Although virtual machines appear to their users to be executing instructions, it is the real CPU that is actually doing the work.

The real CPU uses a technique called time slicing to provide the effect of multiple virtual CPUs. Each virtual machine periodically gains access to the real CPU for a small amount of time called a "time slice." To determine how frequently and for how much time a virtual machine should gain access to the real CPU, CP examines the number of console requests or terminal interrupts the virtual machine has issued during its past time slices. If these were many, CP defines the virtual machine as a conversational user and assigns it the smaller of two possible time slices. If they were few, the virtual machine is a nonconversational user and is assigned the larger time slice. Also, CP gives conversational users more frequent access to the real CPU; nonconversational users get time slices less frequently.

CP allows a virtual machine to gain access to the real CPU only if the virtual machine is not waiting for some resource or activity, such as:

- A page of storage to be brought in.
- An input/output operation to be translated, begun, or completed.
- A CP command to finish executing.

VM/370 includes performance options that can be used to improve the performance of a virtual machine. A virtual machine to which a performance option has been applied is termed "preferred." Although each option could be applied to a different virtual machine, this is not normal practice if optimum performance is required for one or two specific virtual machines.

Two of these options can be used to increase the amount of real CPU time made available to a particular virtual machine: priority and favored execution.

The priority value assigned to a virtual machine is used, in combination with other parameters, to influence the dispatching algorithm and give that virtual machine a larger slice of CPU time, provided it can fully utilize the time. The priority assignment affects the execution of a particular virtual machine as compared with other virtual machines that have the same general execution characteristics. Priority may be assigned by the system operator but is more frequently a parameter of the virtual machine's directory entry.

The favored execution option provides a particular virtual machine an assured percentage of real CPU time, provided it can fully utilize the time. The system operator specifies this option and the percentage by the SET FAVORED command. Only one virtual machine at a time can have this form of the favored execution option.

Another form of the SET FAVORED command, with no percentage specified, can be issued for several virtual machines, to ensure that they gain access to the real CPU more frequently than other virtual machines.

More detailed information on these and other options that improve virtual machine performance is contained in the _VM/370: System Programmer's Guide_.

## Virtual Machine Storage Management

Each virtual machine has storage associated with it. The amount of storage is defined in the VM/370 directory. Each virtual machine functions as if it has a large amount of real storage. However, each virtual machine's storage is created and controlled by CP as virtual storage. The virtual machine's storage can be larger or smaller than the storage of the real machine.

The directory entry contains two sizes for each virtual machine: its normal size and a maximum size. The normal size must be at least 8K (8192) bytes. The maximum size must be no larger than 16 million bytes. Both sizes must be multiples of 4K (4096). The virtual machine usually uses the amount of storage defined as its normal size. However, the user can temporarily redefine his virtual storage size to any value that is a multiple of 4K (4096) and not greater than his maximum size.

Storage in the virtual machine is logically divided into 64K (65,536) byte areas called segments. These are logically divided further into 4K byte areas called pages. For each virtual machine, CP creates and maintains a set of segment and page tables to describe the virtual storage and to reflect the allocation of the virtual storage pages to page frames in real storage. These tables are used by the Dynamic Address Translation feature during virtual machine execution to locate the real storage addresses to which the virtual storage addresses actually refer.

The storage of the real System/370 is physically and logically divided into 4K byte areas called page frames. When a page of virtual storage is brought into real storage, it fits exactly into a page frame.

The heavily used portions of VM/370 are kept in real storage. However, only frequently referenced virtual storage pages are kept in real storage, thus optimizing real storage utilization. A page can be brought into any available page frame; the necessary relocation is done during program execution by CP using dynamic address translation. The active pages from all logged-on virtual machines and from the pageable routines of VM/370 compete for available page frames. When the number of page frames available for allocation falls below a threshold value, CP determines which virtual storage pages currently allocated to real storage are relatively inactive and initiates suitable page-out operations for them.

Inactive pages are maintained on a direct access storage device. If an inactive page has been changed at some time during virtual machine execution, CP assigns it to a paging device, selecting the fastest such device with available space. If the page has not changed, it remains allocated in its original direct access location and is paged into real storage from there the next time the virtual machine refers to that page. A virtual machine program can use the DIAGNOSE instruction to communicate to CP that the information from a specific page(s) of virtual storage is no longer needed; CP then releases the areas of the paging device(s) that had been assigned to hold the specified page(s).

Paging is done on demand by CP. This means that a page of virtual storage is not read (paged) from the paging device to a real storage page frame until it is actually needed for virtual machine execution. No attempt is made by CP to anticipate what pages might be required by a virtual machine. While a paging operation is being performed for one virtual machine, another virtual machine can be executing. Paging operations are initiated and performed by CP and require no action by the virtual machine.

The operating system controlling a virtual machine may execute in extended control mode. This means that it can create and control virtual storage of its own, in addition to the virtual storage it has which is controlled by CP. The virtual machine operating systems that can do this are: OS/VS1, OS/VS2, DOS/VS, and VM/370. (VM/370 can create several virtual storages at once.) In the following example, OS/VS1 is used to illustrate how an operating system handles the virtual storage it creates, and how this is different from the virtual storage that VM/370 creates for a virtual machine.

OS/VS1 creates and controls a single virtual storage. It maintains a set of page and segment tables that relate this virtual storage to the virtual storage of the virtual machine. In VM/370, the terms "first level storage", "second level storage", and "third level storage" refer to real storage, the storage of the virtual machine, and the virtual storage created and controlled by the virtual machine, respectively. When OS/VS1 is executing, instructions and data from third level storage must be available to the CPU. For this purpose the real machine cannot use the page tables created by OS/VS1 nor the page tables created by CP. The real machine must have a set of page and segment tables that relate third level storage to first level storage. CP dynamically constructs and maintains such tables, called shadow tables. CP maintains a single set of shadow page tables for any one virtual machine. A single set is all that is necessary for OS/VS1, OS/VS2, or DOS/VS.

However, when VM/370 itself is used as a virtual machine operating system, it can create multiple virtual machines, each with its own virtual storage. In this case, the shadow tables are invalidated by CP whenever it passes control from one virtual machine to another.

One or more segments of virtual storage can be shared among virtual machines. The information to be shared must be read-only; it may be data or reentrant program modules. The information to be shared must be part of a monitor or operating system (for example, CMS) that has been recorded or "saved" on a VM/370 system disk. Storage segments cannot be shared among virtual machines executing with the Virtual Machine Assist feature on, or in extended control mode with translate on.

CP provides three performance options to reduce or eliminate paging requirements of specific virtual machines: locked pages, reserved page frames, and virtual=real.

The LOCK command can be used by the system operator to lock specific user pages of virtual storage into real storage. This eliminates paging activity for these pages. Since this option reduces the number of page frames that are available for use by other virtual machines, only frequently used pages should be locked in real storage.

A more flexible approach than locked pages is reserved page frames. The system operator assigns to a specified virtual machine a certain number of page frames for its own use. Pages are not locked into these page frames; they can be paged out, but only for other active pages of the same virtual machine. This option is usually more efficient than locked pages, since the pages that remain in real storage are those that are most active at the moment, as determined by CP. Although several virtual machines can have locked pages, only one virtual machine at a time can have reserved page frames.

During VM/370 system generation, the installation can give one or more virtual machines the virtual=real option. However, only one virtual=real machine can be active at any one time. With this option, the virtual machine's storage is allocated directly from real storage at the time VM/370 is initially loaded, and remains so allocated unless released by an operator command. All pages except the virtual machine's page zero are allocated to the corresponding real storage locations. (In order to control the real computing system, real page zero must be controlled by CP.) Consequently, the real storage size must be large enough to accommodate the CP nucleus, the entire virtual=real machine, and the remaining pageable storage of VM/370 and the other virtual machines.

The virtual=real option improves performance in the selected virtual machine because CP no longer has to perform paging operations for it. The performance of other virtual machines may be adversely affected unless enough real storage is available for their paging requirements, so care should be taken in assigning the virtual=real storage size. The _VM/370: Planning and System Generation Guide_ discusses some situations in which the virtual=real option is necessary.

Figure 8 illustrates real storage allocation for both a normal DOS batch virtual machine, and one defined with the virtual=real option.

## Virtual Machine I/O Management

The virtual machine operating system is responsible for the operation of all virtual devices associated with it. These virtual devices may be defined in the directory entry of the virtual machine, or they may be attached to (or detached from) the virtual machine's configuration while it remains logged on. Virtual devices may be dedicated, as when mapped to a fully equivalent real device; shared, as when mapped to a minidisk or when specified as a shared virtual device; or spooled by CP to intermediate direct access storage.

In a real machine running under control of OS, input/output operations are normally initiated when a problem program requests OS to issue a START I/O instruction to a specific device. Device error recovery is handled by the operating system. In a virtual machine, OS can perform these same functions, but the device address specified and the storage locations referenced will both be virtual. CP translates the virtual specifications to real.

Because the virtual machine executes only in virtual (not real) supervisor state, CP gains control when the START I/O instruction is issued by the virtual machine operating system. CP copies into its own work area the channel command list specified by the operating system, and pages into real storage all virtual storage

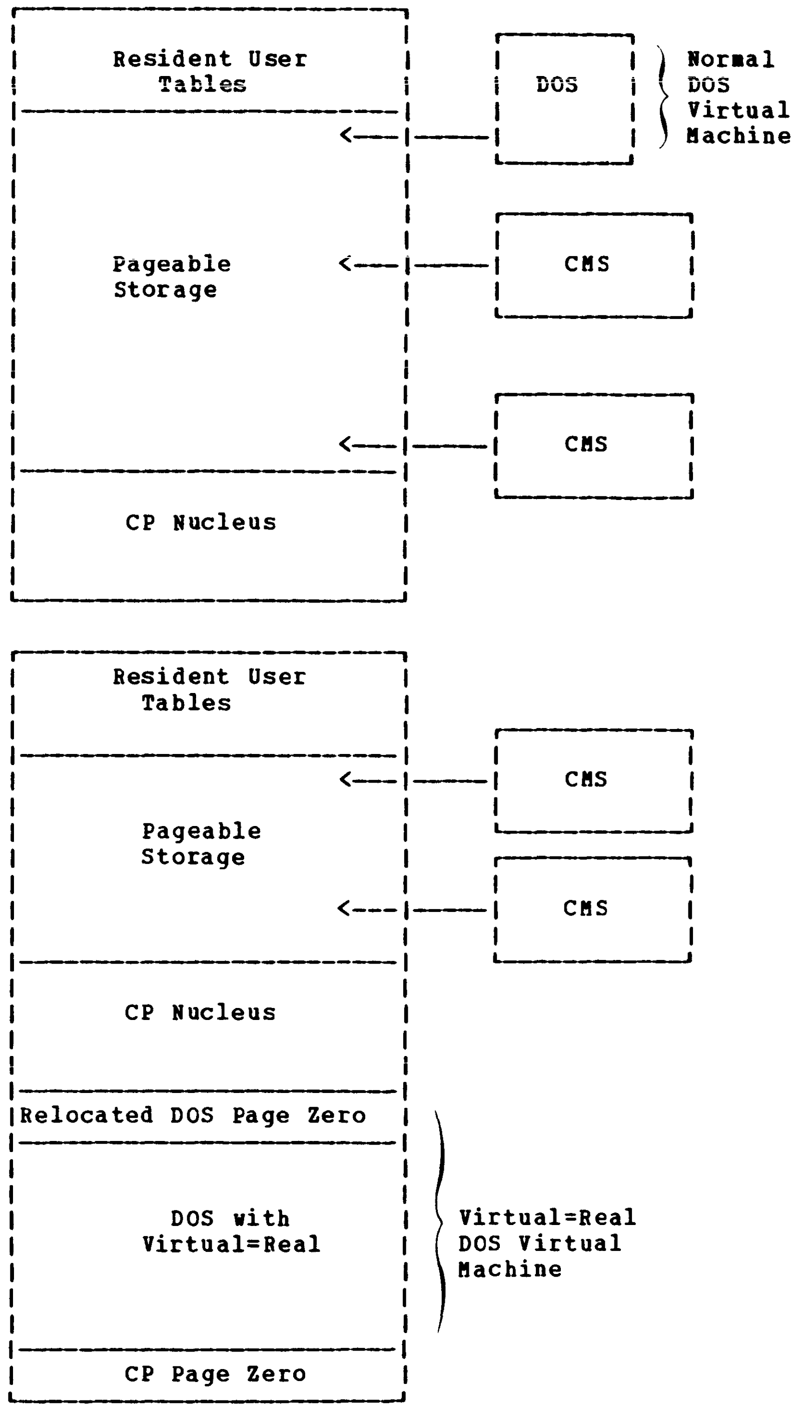


Figure 8. Normal and Virtual=Real Virtual Machines

locations required for data transfer. The specified pages are fixed in real storage until completion of the input/output operation. If a single channel command word has specified a data area extending over multiple pages of contiguous virtual storage, CP generates channel programs that use channel indirect data addressing to handle discontiguous page frames. If the virtual device is a minidisk, any cylinder numbers specified are modified to reflect the true location of the data. The virtual device address is mapped to the real device and an actual input/output operation is scheduled.

During this processing, the virtual machine has been marked as not executable by CP. When the virtual machine gains control it is given a suitable condition code (as on a real machine) to indicate the status of the START I/O operation. In addition, the interrupts caused by the input/output operation are reflected to the virtual machine for its interpretation and processing.

If input/output errors occur, CP does not initiate error recovery operations; these are the responsibility of the virtual machine operating system. Basic error recording is, however, provided by CP. For more information on error processing, see the "Reliability, Availability, and Serviceability" section and "Appendix A: Recovery Management Support".

The programs to be run in a virtual machine (except a virtual=real machine) generally must not include dynamically modified channel programs. These and other restrictions that apply to virtual machines are discussed in the _VM/370: Planning and System Generation Guide_.

A virtual disk device can be shared among multiple virtual machines. Virtual device sharing is specified in the directory entry or by a user command. If it is the latter, the user must supply an appropriate password to the system before gaining access to the virtual device.

A particular virtual machine may be assigned read-only or read/write access to a shared disk device. CP verifies each virtual machine input/output operation against the parameters in the virtual machine configuration to ensure device integrity.

Virtual disk devices may be defined for temporary use by a virtual machine. In that case, CP allocates real disk storage to the virtual machine until the virtual machine logs off or specifically detaches the temporary virtual device.

A virtual machine may be assigned a dedicated channel, via the ATTACH CHANNEL command. A virtual machine assigned a dedicated channel has that channel and all of its devices for dedicated use. CP translates the virtual storage locations specified in channel commands to real locations and performs any necessary paging operations, but does not need to translate any device addresses. The virtual devices on a dedicated channel must have direct, real equivalents (for example, minidisks are not allowed), and the virtual and real device addresses must be identical. A channel dedicated to a virtual machine cannot be used. Virtual machines may have a mixture of dedicated and nondedicated channels.

The initiation of a virtual input/output operation by CP can be simplified through use of the Diagnose interface by the virtual machine operating system. The Conversational Monitor System, which has been designed specifically for the virtual machine environment, uses this interface instead of the normal START I/O instruction for most of its input/output operations. When Diagnose is used, CP handles input/output error recovery operations.

Input/output operations initiated by CP for its own purposes, for example, paging and spooling, are performed directly and are not subject to the translation process described in the preceding paragraphs.

## Spooling Unit Record I/O

CP spooling facilities allow multiple virtual machines to share unit record devices. Since virtual machines controlled by CMS ordinarily have low requirements for unit record input/output, such device sharing is advantageous, and it is the standard mode of system operation.

CP, not the virtual machine, controls the unit record devices that have been designated as spooled in the directory entry. When the virtual machine issues a START I/O instruction to a spooled unit record device, CP intercepts the instruction and modifies it. CP moves the data into page-size records (that is, 4096 byte blocks) on a VM/370 disk area that serves as intermediate storage between the real unit record device and the virtual machine.

Input spool files, that is, data available at a virtual card reader, can be created when the real machine operator feeds into the real card reader, cards that are preceded by a USERID card specifying to which virtual machine the input is to go. Input spool files can also be created by using an IBM 2780 Data Transmission Terminal. CP provides a virtual machine monitor program that transfers files between a 2780 and the devices of any specified virtual machine. In this way, cards at a remote location can be made available to any virtual machine.

Output spool files are created on direct access storage when the virtual machine operating system writes to a virtual punch or printer. Real output is scheduled for a real printer or punch, or for remote output, whenever a user logs off the system or issues a CP spooling command to close the file.

If the direct access storage space assigned to spooling is filled, spooling will stop and the virtual unit record devices will appear not ready. The spooling operator may make additional spooling space available by purging existing spool files or by assigning additional direct access storage space to the spooling function.

Specific files can be transferred from the spooled card punch or printer of a virtual machine to the card reader of the same or another virtual machine. (A virtual card reader is not limited to 80 characters.) Files transferred between virtual unit record devices by the spooling routines are not physically punched or printed. With this method, files can be made available to multiple virtual machines, or to different operating systems executing at different times in the same virtual machine.

CP spooling options include printing multiple copies of a single spool file, backspacing any number of printer pages, and defining spooling classes for the scheduling of real output.

## Spooling Virtual Console I/O

CP allows the user to spool his virtual machine's console input/input on disk, instead of, or in addition to, having it displayed at his terminal. The data spooled includes messages from or to the virtual machine operating system, CP commands entered by the user, CP messages and responses, and messages from or to the system operator. This facility, invoked by the SPOOL CONSOLE command, is particularly useful when the virtual machine is executing with the terminal disconnected, since the virtual console output, which would otherwise be lost, is saved on disk. The saved data is later printed on the real printer.

## Remote Spooling

CP, in conjunction with RSCS, provides support for remote spooling, that is, RSCS provides for transmission of files across a teleprocessing network controlled by the RSCS control program. The section "Remote Spooling Communications Subsystem" provides details about the subsystem and how it is used.

## CP Commands

CP commands are used interactively to control the real computing system and VM/370, and to provide user control of virtual machines and associated control program facilities.

CP commands can be used at any time, without regard to which operating system is controlling the user's virtual machine. To issue CP commands, the user must first suspend execution in the virtual machine by presenting an attention interrupt to VM/370's control program; this is the virtual machine equivalent of pressing the stop button on a real computing system. However, the CMS user can issue CP commands without leaving the CMS environment, that is, without presenting an attention interrupt.

### PRIVILEGE CLASSES

Each user of VM/370 is assigned one or more privilege classes as part of the directory entry of his virtual machine. A user's privilege class(es) defines his allowable subset of CP commands.

Figure 9 identifies the privilege classes, their functions, and the users to whom they are assigned, and names the publications in which the commands for each class are described.

| Class | User and Function |
|---|---|
| A[1] | Primary System Operator: The class A user controls the VM/370 system. Class A is assigned to the user at the VM/370 system console at IPL time. The primary system operator is responsible for the availability of the VM/370 system and its communication lines and resources. In addition, the class A user controls system accounting, broadcast messages, virtual machine performance options and other command operands that affect the overall performance of VM/370.<br>Note: The class A system operator who is automatically logged on during CP initialization is designated as the Primary System Operator. |
| B[1] | System Resource Operator: The class B user controls all the real resources of the VM/370 system, except those controlled by the primary system operator and spooling operator. |
| C[1,2] | System Programmer: The class C user updates certain functions of the VM/370 system. |
| D[1] | Spooling Operator: The class D user controls spool data files and specific functions of the system's unit record equipment. |
| E[1,2] | System Analyst: The class E user examines and saves certain data in the VM/370 storage area. |
| F[1,3] | Service Representative: The class F user obtains and examines in detail, certain data about input and output devices connected to the VM/370 system. |
| G[4] | General User: The class G user controls functions associated with the execution of his virtual machine. |
| Any[1,4] | Any User: The class Any user has limited use of VM/370 to gain initial access to the VM/370 system. |
| H | Reserved for IBM use. |

[1]Described in the VM/370: Operator's Guide.
[2]Described in the VM/370: System Programmer's Guide.
[3]Described in the VM/370: OLTSEP and Error Recording Guide.
[4]Described in the VM/370: Command Language Guide for General Users.

Figure 9. CP Privilege Class Descriptions

## GENERAL USERS

In order to become active in the system, the user prepares his terminal and establishes a line connection to the real computing system that is running VM/370. At that point, the LOGON command identifies the user to VM/370, which then creates the control blocks necessary to simulate the virtual machine configured in his directory entry. Or the user's terminal can be a remote terminal for a multiple-access virtual machine operating system (such as APL\DOS-360). To identify himself to a multiple-access system, the user issues the DIAL command and is, thereafter, controlled by the multiple-access system directly. The user's terminal must be of a type supported by the multiple-access system.

Ordinarily, the user immediately loads an operating system into his virtual machine, using the IPL command. Execution in the virtual machine can be stopped at any time, and the user can then invoke commands to simulate the functions of the operator's console or otherwise to control his virtual machine. Some of these commands are:

| Command | Meaning |
|---|---|
| EXTERNAL | Causes an external interrupt. |
| ADSTOP | Defines an instruction address stop location in virtual storage. |
| DISPLAY | Displays specified virtual machine registers or virtual storage contents in hexadecimal or EBCDIC. |
| DETACH | Removes a specified device from the virtual machine configuration. |
| READY | Simulates a device end interrupt from a virtual device. |
| SPOOL | Alters the spooling control options (such as number of copies) for one or more virtual unit record devices that are controlled by the spooling facilities. In addition, this command transfers data files among users and remote stations, and starts and stops console spooling. |
| STORE | Inserts data into virtual machine registers or virtual storage. |
| BEGIN | Resumes execution in the virtual machine (the functional equivalent of pressing the Start key on a real computing system). |
| LINK | Makes a specified virtual direct access storage device a part of the virtual machine configuration if the device is defined as sharable and the user can supply the appropriate password. |
| QUERY | Interrogates certain system information such as the log message, the number of spool files, or the virtual machine configuration. |
| SET | Establishes certain system values such as the level of error message to be printed or the amount of VM/370 editing for terminal input lines. |
| TERMINAL | Allows the user to define the VM/370 logical editing characters and the logical linesize of I/O to and from his terminal. |

OTHER USERS

The following discusses some of the functions that other VM/370 users may perform. VM/370 system operators can use the SET command to dynamically provide any of the VM/370 performance options, except virtual=real, to a particular virtual machine. (The virtual=real option is defined in the VM/370 system generation and specified in the directory entry of the selected virtual machine.) System operators control the orderly activity of the real computing system with FORCE to terminate a particular virtual machine, SHUTDOWN to terminate all VM/370 functions such that a VM/370 restart can be performed, and ACNT to create accounting records for all active users. System operators can define a paging characteristic in SET that will affect the amount of multiprogramming attempted by VM/370.

System resource operators can control communication lines with ENABLE and DISABLE, logically remove a device from the real computing system with DETACH and VARY, and add a device to either the real computing system or a virtual machine configuration with ATTACH.

Spooling operator commands include BACKSPAC, FLUSH, PURGE, REPEAT, SPACE, HOLD, and FREE.

System analysts can issue DCP to display real storage locations at the terminal or DMCP to print them offline. They can use QUERY to learn current system paging characteristics. SAVESYS can be issued to save a "core image" copy of a virtual machine's storage space, registers, and program status word on the VM/370 system residence disk.

## Performance

The performance of any computing system is judged by how efficiently and quickly it processes the work it has to do.

The following factors influence the performance of a VM/370 system:

- The System/370 model used.
- The total number of virtual machines executing.
- The type of operating systems being used in the virtual machines.

- The type of work being done by each virtual machine.
- The type, capacity, and number of the primary paging devices.
- The number of channels available.
- The channel operating mode, block multiplexer or selector.
- The amount of real storage available.
- The use of the Virtual Machine Assist feature.

In general, the best performance is seen in a VM/370 system that has a large amount of real storage, many channels available, many large primary paging devices, and relatively few virtual machines executing at any one time. However, virtual machines that are executing CMS make smaller demands on system resources than do virtual machines running other operating systems, particularly those that manage their own virtual storage. Therefore, a VM/370 system can satisfactorily manage many more CMS machines than, for example, OS machines.

The performance of an operating system such as OS can be improved by substituting CP paging for virtual machine I/O operations. This is done by specifying as resident as many frequently-used OS functions as possible (for example, transient subroutines and ISAM indexes). CP can then bring the page containing the routine or data into real storage when it is needed.

Instead of constructing problem programs with complicated overlays, programmers should allow CP's paging facilities to manage storage dynamically.

## VIRTUAL MACHINE CHANNEL MODE SELECTION

Virtual machine SIO operations are simulated by CP in three ways: byte-multiplexer, selector, and block multiplexer channel mode.

Virtual byte-multiplexer mode is reserved for I/O operations that apply to devices allocated to channel zero.

Selector channel mode, the default mode, is the mode of operation for any channel that has an attached Channel-to-Channel Adapter (CTCA), regardless of the selected channel mode setting (the CTCA is treated as a shared control unit and, therefore, it must be connected to a selector channel).

Block multiplexer channel mode is a CP simulation of real block multiplexer operation; it allows the virtual machine's operating system to overlap SIO requests to multiple devices connected to the same channel.

Note: CP simulation of block multiplexing does not reflect channel available interrupts (CAIs) to the user's virtual machine.

The selection of block multiplexer channel mode or selector channel mode is effective regardless of the real channel devices on the System/370. The channel operating mode is selected via the CP DEFINE command (see the *VM/370: Command Language Guide for General Users*) or via a Directory service program option (see the *VM/370: Operator's Guide*).

## VIRTUAL MACHINE ASSIST FEATURE

The Virtual Machine Assist feature, which improves the performance of VM/370, is a combination of a CPU feature and VM/370 programming. Virtual storage operating systems (such as OS/VS1 and DOS/VS) that run in problem state under control of VM/370, use many privileged instructions and SVCs that cause interrupts that VM/370 must handle. When the Virtual Machine Assist feature is used, many of these interrupts are intercepted and handled by the CPU; consequently, VM/370 performance is improved.

The Virtual Machine Assist feature is available with System/370 Models 135, 145, 158, and 168.

Whenever VM/370 is IPLed on a CPU that has the Virtual Machine Assist feature, the feature is enabled for all virtual machines on the system. The system operator can disable and enable the feature for the system, using the SET command.

When a user logs on, the Assist feature is enabled for his virtual machine, if it is enabled for the system. The general user can set the feature off for his virtual machine, and later set it on again. He can also control whether SVC interrupts are handled by the Assist feature or by VM/370.

More information about SVC handling appears under "Error Recording Procedures" in "Appendix A: Recovery Management Support."

Under some conditions, the Virtual Machine Assist feature cannot be used. CP automatically turns the feature off if the user invokes certain TRACE functions, attaches a dedicated channel, or attempts to load a system containing a shared segment. CP automatically turns the feature on again when the user ends the TRACE, detaches the dedicated channel, or loads a system that does not contain a shared segment.

For further information about SVC handling, the Assist feature, and improving performance in the virtual machine environment, refer to the VM/370: System Programmer's Guide.

### PERFORMANCE MEASUREMENT AND ANALYSIS

The VM/370 control program has two commands that measure system performance: MONITOR and INDICATE. The MONITOR command collects system measurement data offline for the system operator or system analyst, while the INDICATE command displays system measurement data online for the system analyst or general user.

The MONITOR command gathers data relating to most aspects of system performance and writes the data on tape. When the data collected on tape is summarized, it may indicate the conditions contributing to performance degradation.

The INDICATE command displays, at the terminal, some key information about the system to show the current performance conditions. INDICATE displays the system conditions existing at the time it is issued. If, after using the INDICATE command, the system analyst wants more extensive data collection and reduction he can use the MONITOR command.

Refer to the VM/370: System Programmer's Guide, the VM/370: Operator's Guide, and the VM/370: Command Language Guide for General Users for details about the MONITOR and INDICATE commands.

## VM/VS Handshaking

VM/VS Handshaking is a communication path between the VM/370 control program and OS/VS1 that makes each system control program aware of any capabilities or requirements of the other. VM/VS Handshaking consists of:

- Closing CP spool files
- Processing VS1 pseudo page faults
- Providing an optional nonpaging mode for VS1 when it is run with VM/370
- Providing miscellaneous enhancements for VS1 when it is run under the control of VM/370

### CLOSING CP SPOOL FILES

When the handshaking feature is active, VS1 closes its CP spool files when VS1 job output from its Direct System Output (DSO), terminator, and output writers is complete. Once the spool files are closed, they are processed by VM/370 and sent to the real printer or punch. With the VM/VS Handshaking feature, virtual machine operator intervention is not required to close these spool files.

### PSEUDO PAGE FAULTS

A page fault is a program interruption that occurs when a page that is marked "not in storage" is referred to by an instruction within an active page. The virtual machine operating system referring to the page is placed in a wait state while the page is brought into real storage. Without the handshaking feature, the entire VS1 virtual machine is placed in page wait by VM/370 until the needed page is available.

However, with the handshaking feature, a multiprogramming (or multitasking) VS1 virtual machine can dispatch one task while waiting for a page request to be answered for another task. VM/370 passes a pseudo page fault (program interruption X'14') to VS1. When VS1 recognizes the pseudo page fault, it places only the task waiting for the page in page wait, and can dispatch any other VS1 task. Thus, when VS1 uses pseudo page faults, its execution under the control of VM/370 more closely resembles its execution on a real machine.

### VS1 NONPAGING MODE

When VS1 is run under the control of VM/370 and its virtual address space is equal to the size of the VM/370 virtual machine, VS1 is in nonpaging mode. When VS1 executes in nonpaging mode, it uses fewer privileged instructions and avoids duplicate paging.

However, VS1 may have a larger working set when it is run in nonpaging mode than when it is not. Nonpaging mode is available only when the VM/VS Handshaking feature is active.

MISCELLANEOUS ENHANCEMENTS

When OS/VS1 is run in a VM/370 environment, a certain amount of duplication results. However, when the handshaking feature is active, the VS1 virtual machine avoids some of those instructions or procedures that would result in inefficiency. For example, VS1 avoids:

- ISK (Insert Storage Key) instructions and uses a key table instead
- Seek separation for 2314 direct access devices
- The ENABLE/DISABLE sequence in the VS1 I/O Supervisor (IOS)
- TCH (Test Channel) instructions preceding SIO (Start I/O) instructions

VM/VS HANDSHAKING REQUIREMENTS

VS1 must generated with the VM/370 option and run with a version of VM/370 that supports VM/VS Handshaking. In addition, VS1 must be run in Extended Control (EC) mode.

When VM/VS Handshaking is available in an OS/VS1 virtual machine, the pseudo page fault handling portion of handshaking is not available until the operator of the virtual machine issues the CP SET PAGEX ON command. The pseudo page fault portion of the VM/VS Handshaking feature can be set on and off with the CP SET command. The CP SET command is described in the _VM/370: Command Language Guide for General Users_; more information about using the VM/VS Handshaking feature can be found in the _VM/370: System Programmer's Guide_.

## Reliability, Availability, and Serviceability

VM/370 provides increased reliability, availability, and serviceability (RAS) to its users in two ways:

1. Through the design of the environment of multiple virtual machines. This is discussed below.
2. Through use of the RAS features of the System/370 hardware and VM/370 system control programming. These are discussed in detail in "Appendix A: Recovery Management Support."

RAS FEATURES OF VM/370 DESIGN

In the virtual machine environment, each user is effectively isolated from the activities of all other virtual machine users.

1. CP isolates the virtual machine storage by referring to it via its own page and segment tables only; a virtual machine cannot generate a storage address that refers to any storage area except its own.
2. Passwords must be used to gain access to the system and to shared disk files.
3. The data on shared or critical disk files can be protected by designating it as "read-only".
4. The termination of a virtual machine affects only that machine and its user (or users, if it is a multiple-access system). The user is sent an appropriate message at the virtual operator's console before the termination. The line will be left open, but the user will need to log on to VM/370 and load his operating system again.
5. Users can run concurrently as many versions, levels, types, and copies of operating systems as they require. Systems can be generated, tested and modified in one virtual machine while production work is being run on another. With no need for a dedicated machine, a new system can be fully tested before conversion with no impact on the production work schedule.
6. To assist in locating programming malfunctions, VM/370 provides commands to trace, examine, and alter the operation of a virtual machine and to examine and alter the contents of its virtual storage.
7. Using the Online Test Standalone Executive Program (OLTSEP), a service

representative can perform online diagnosis of I/O errors for most devices attached to the System/370. He can run OLTSEP in a virtual machine while other work is being done on other virtual machines.

# Conversational Monitor System

The Conversational Monitor System (CMS) is the major subsystem of VM/370. Together with the control program of VM/370, it provides a time sharing system suitable for direct problem solving and program development. CMS is an operating system that executes only in a VM/370 virtual machine. (CMS uses the Diagnose interface for all of its disk and tape input/output operations and includes no error recovery routines.)

CMS is a conversational, single user system. The user's interface to CMS is the virtual operator's console, that is, the terminal used to gain access to VM/370.

CMS has no multiprogramming capabilities, as it is designed to run in a VM/370 virtual machine. CP provides the time sharing environment; CMS provides the conversational user interface. Using CMS, the user may write programs to be executed under CMS or under another virtual machine operating system.

CMS is an extension of the Cambridge Monitor System, a major component of the predecessor of VM/370, CP-67/CMS. "Appendix E: Compatibility of VM/370 with CP-67/CMS" discusses the relationship between these systems.

## CMS Configuration

A virtual machine that is to use CMS is configured much the same as any other virtual machine, with a few special considerations.

The CMS virtual machine must be assigned at least 320K bytes of virtual storage, of which 128K is used by the CMS nucleus. User programs that execute in CMS may increase this requirement. The virtual storage size may be defined as large as 16 million bytes, in multiples of 4K.

The most active portion of the nucleus can be shared by users of CMS via the shared segment facilities of VM/370. The amount shared is 64K, one full segment. This allocation is not locked in real storage, which means that a particular shared page of it may not be in real storage at any given time. However, the most active pages tend to remain in real storage.

CMS supports unit record devices only if they are virtual and use the CP spooling facilities. Real unit record devices cannot be dedicated to the CMS machine since CMS includes no unit record error recovery procedures.

CMS supports tape devices, but disk volumes are the primary external storage for CMS command processing.

Each CMS user is assigned at least two disk volumes. (These are virtual disks, usually minidisks.) These are:

1. A read-only system disk volume, containing the CMS nucleus, disk resident CMS commands, and system library. This volume is shared among all CMS users.

2. A read/write disk for permanent and temporary user files. This user disk can be defined as any size required for the user files, as long as it is a multiple of one cylinder and is no larger than one volume. (On a 3330 device, the user disk must be no larger than 246 cylinders; on a 70-megabyte 3340 device, a maximum of 682 logical cylinders can be used.)

Figure 10 shows a virtual machine configured to run CMS. For the minimum CMS configuration, the virtual tapes would not be included.

Figure 11 shows the VM/370 directory entry for the CMS virtual machine shown in Figure 10. It defines two virtual disks and the required spooled unit record devices. The tapes are not defined in the directory entry but are attached to the virtual machine by the system operator when needed.

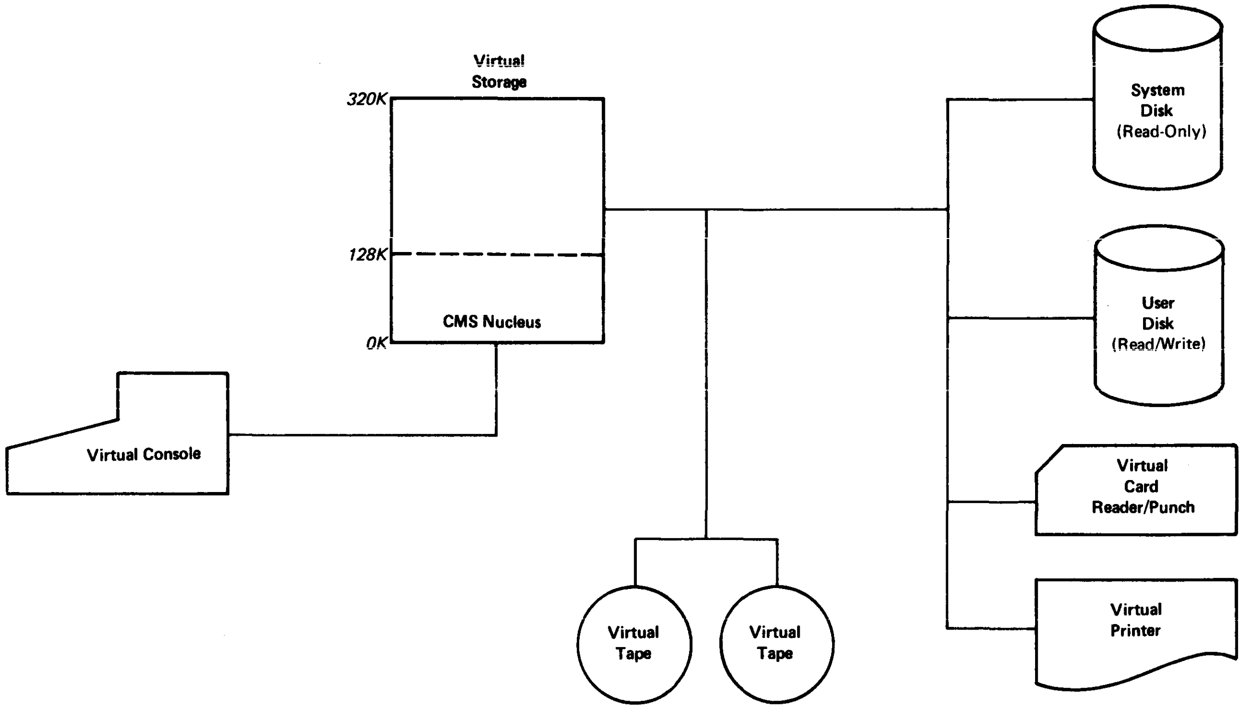


Figure 10. Typical CMS Configuration

The first virtual disk shown in Figure 11 is defined by the LINK entry to share the CMS system on a read-only basis; the CMS system disk is owned by the user CMSSYS (usually the system programmer) and is assigned the virtual device address 190 in both the CMSSYS and the SMITH virtual machine configurations. The second virtual disk defined in Figure 11 is owned by the user SMITH. It has virtual address 191, is a minidisk located on the real volume labeled CMSVL1, and occupies 5 cylinders starting with real cylinder 025 on that volume.

```
USER SMITH JOHN
ACCOUNT 5976
CONSOLE 009 3215
SPOOL 00C 2540 READER
SPOOL 00D 2540 PUNCH B
SPOOL 00E 1403 A
LINK CMSSYS 190 190 R
MDISK 191 2314 025 005 CMSVL1 W
```

Figure 11. Directory Entry for a CMS Machine

## CMS File System

CMS supports 2314, 2319, 3330, 3333, and 3340 disks. The tracks of a CMS disk are preformatted by the CMS FORMAT command into 800-byte blocks. Using the blocks, the CMS file system is able to present the user with the effect of logical fixed or variable length records, and provide sequential or direct access to files.

CMS provides special files containing macro libraries and program libraries, and special commands to use and modify them. The user or installation can create additional macro and program libraries if needed.

CMS requires the system residence volume to be online. Each user may have up to nine virtual disks online at any one time. (All nine of these might reside on one real disk.)

The user disks are differentiated by a filemode designator, assigned when the disk is made active. The filemode consists of a letter and a number. The number indicates the disk's access mode, as explained below. The letter, which may be A through G, S, Y, or Z, allows the user to define a standard order-of-search for disk files. S denotes the System disk.

Each virtual disk may be defined as read-only or read/write, and may be shared among users as described in the "Virtual Machine I/O Management Section."

User files in CMS are identified with a fileid consisting of three designators: filename, filetype, and filemode. The filename is the name the user assigns to the file. The filetype may specify particular characteristics of the file. For example, the filetype EXEC indicates that the file is an EXEC procedure; ASSEMBLE indicates that the file consists of assembler language source statements.

The filemode describes the location and access mode of the file. The letter (A through G, S, Y, or Z) indicates that disk in whose directory the file is listed. The numeric part of the filemode defines the type of access permitted to the file. It indicates whether the file is private, read/write, read-only, or in simulated OS format. The filemode works in conjunction with the access status of the disk. For example, a user can gain access to a file with the "private" filemode only if the disk on which it is stored is in read/write mode. Any user sharing the disk, and having read-only access to the disk, is unable to gain access to files on the disk marked "private." The filemode designator can thus be used to provide limited data security.

A single user file may contain up to 12,848,000 bytes of data grouped into up to 65,535 logical records, all of which must be on a single virtual disk. The maximum number of files per real disk is 3400 for a 3330, 3333, or 3340 disk, or 3500 for a 2314 or 2319.

CMS disk files are written as 800-byte records, which usually are not physically contiguous on the disk. They are allocated and deallocated automatically by CMS as the file size demands. Each virtual disk contains a CMS file directory, which gives format and size information for each file on the virtual disk and includes a pointer to the file's chain link records. (The CMS file directory, or a specified subset of it, is brought into virtual storage when the disk is made available to the CMS user; it is updated at least once per command if the status of any file on that virtual disk has been changed.) Each file's chain link records point to the 800-byte records of that file.

Figure 12 illustrates the CMS file structure. The CMS file directory entry for the file named PROG1, filetype COBOL, points to the chain link records for that file, each of which points to a separate 800-byte record of the file.

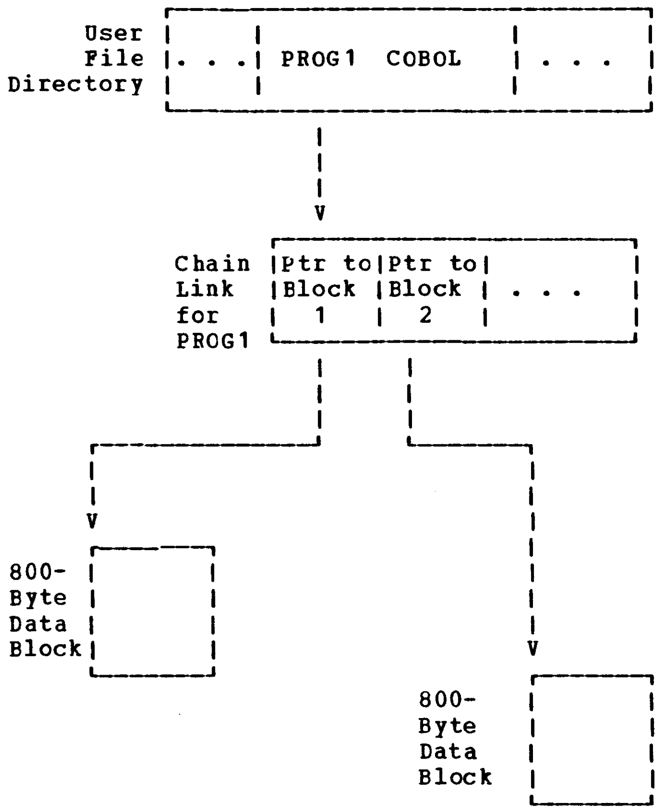


Figure 12. CMS File Format

CMS automatically opens and closes all accessed files (including spool files) for each command or user program executed. Specific files may be spooled between virtual machines to accomplish file transfer between users. Service commands allow such file manipulations as writing an entire disk or specific disk file(s) to a tape, printer, punch, or the terminal. Other commands transfer data from a tape or virtual card reader to disk, rename files, copy files, and erase files. CMS files can be written onto and restored from unlabeled tapes via CMS service commands. Tape labels are not supported by CMS.

CMS dynamically allocates compiler work files at the beginning of command execution on whichever active user disk has the most available space (although their location may be specified by the user), and deallocates them at completion. Compiler object decks and listing files are normally allocated on the same disk as the input source file or on the primary read/write disk. They are identified by the input filename together with the filetype TEXT or LISTING.

In addition to reading and writing CMS files, CMS can read sequential DOS files and sequential and partitioned OS data sets. The CMS MOVEFILE command and the OS BSAM, BPAM, and QSAM macros can be used to read OS and DOS data. CMS cannot, however, write or update OS data sets or DOS files.

Problem programs that execute in CMS can create files on unlabeled tape in any record and block size; the record format can be fixed, variable, or undefined.

## Initialization and Dump/Restore

The OS IBCDASDI service program, which replaces the MINIDASD service program, initializes all types of real and virtual minidisks supported by VM/370. Details on how to use this program are in the _VM/370: Operator's Guide_, and the _IBM System/360 Operating System: Utilities_, Order No. GC28-6586-13.

The CMS FORMAT command initializes minidisks for CMS.

A CP Format program formats CP-owned volumes, such as the system residence, paging, and spooling disks.

The DDR service program of VM/370, which executes under CMS but provides service to all system users, dumps and restores all types of minidisks.

## CMS Command Language

The CMS command language is flexible and can be tailored in the following ways by the installation or by individual users.

Most CMS commands can be entered by the user in a truncated form (for example, "i" can represent "ipl", "a" can represent "assemble"). CMS maintains an ordered list of command names, from which it determines which command the truncated form represents. The installation can modify the sequence of the command list and the valid limits of truncation.

Each user (or installation) can define synonyms for any or all command names.

Any executable program stored on a CMS system or user volume can be invoked by name as a command in the same way as any other command is invoked, that is, by typing the program name, followed by any required operands, at the terminal.

The EXEC processor of CMS can be used to define new commands that are combinations of existing commands. Such new commands, called EXEC procedures, eliminate the tedious re-keying of frequently used sequences of commands. The EXEC processor has logical capabilities; EXEC procedures can test the contents of variables, branch on specified conditions, and execute programmed loops. A special EXEC procedure called a PROFILE EXEC can be invoked automatically when the user issues his first command in the CMS environment; it initializes that user's virtual machine according to the information in his PROFILE EXEC file. For more information on how to create and use EXEC procedures, see the _VM/370: EXEC User's Guide_.

## Program Development and Execution

The Conversational Monitor System includes a wide range of functions to facilitate program development. These include commands to create and compile source programs, to modify and correct source programs, to build test files, to execute and test programs and to debug from the terminal. The commands of CMS are especially useful for OS program development, but may be used with other operating systems as well.

### MANIPULATION OF USER FILES

Before a user can create a CMS file, he must make a virtual disk active and define its mode with the CMS ACCESS command. This virtual disk must be part of his virtual machine configuration, as defined by his directory entry or by use of the CP LINK or ATTACH command. Before the virtual disk can be used for the first time, it must be initialized by the CMS FORMAT command, which formats the virtual disk into 800-byte blocks as described under "CMS File System."

A user can create a CMS file on disk by using the commands READCARD, DISK, or TAPE. Files can also be created at the terminal, by issuing the EDIT command and then typing in the input.

The FILEDEF command defines the location and characteristics of OS data sets and DOS files that are to be handled by the OS simulation routines.

The COPYFILE command copies a file or combines two or more files according to specifications in the command. The UPDATE command changes a specified file according to a file containing change control records.

The TYPE command displays all or part of a specified CMS file at the terminal. The LISTFILE command displays status information about all or specified subsets of user files. A single active disk, or all active disks, may be specified. The list displayed may contain only fileids (thus showing a user what files exist on his disk), or it may include file format, allocation, and date information.

The RENAME command changes a specified fileid, or some component of it, such as filetype. The ERASE command deletes a file or a group of related files from a user's read/write disk.

## The CMS Editor

The EDIT command and its subcommands comprise the CMS Editor. With the CMS Editor, a user can create a file by typing the data in at the terminal. He can scan all or part of the file, and insert, change, or delete records.

For example, by entering the subcommand LOCATE with all or part of a record in the file, the user can locate the record, as in the following statement:

```
LOCATE /DATA DIVISION/
```

The CMS Editor scans the file until the first occurrence of the specified characters is found. The record containing this data is then displayed at the terminal.

Once the record is located, it can be changed, using the CHANGE subcommand, as in this sequence:

```
LOCATE /IDEMTIFICATION/
CHANGE /IDEMTIFICATION/IDENTIFICATION/
```

In this record, "IDEMTIFICATION" is changed to "IDENTIFICATION". In addition, if the user has issued the EDIT subcommand VERIFY ON, the changed line is automatically displayed at the terminal.

All occurrences of a specific string of characters can be changed with a single command:

```
CHANGE /USER IDENTIFICATION/USERID/ * *
```

Every occurrence of "USER IDENTIFICATION" in this file is changed to "USERID". This is called a "global" change.

If a file is edited at a display terminal (such as the 3277 Display Station) and VERIFY is on, changes made to the file are automatically reflected in the display. For example, a line inserted into the file appears in the display in its proper position in the file. A line that is deleted from the file is removed from the display. Changes made to the current line appear in that line in the display.

Detailed information about the CMS Editor and how to edit a file, including a discussion of editing at display terminals, can be found in the VM/370: EDIT Guide.

A text processor called SCRIPT/370 is available as an IUP (Installed User Program) for use under CMS. (See "Appendix C: Language Processors and Emulators.") SCRIPT/370 includes manuscript facilities that create formatted output from one or more CMS files containing text and/or text-manipulating control words.

## PROGRAM COMPILATION AND EXECUTION COMMANDS

The compilers executable under CMS are invoked by name and provided with a source file whose filetype designator indicates the compiler. The commands include: ASSEMBLE, ASM3705, VSBASIC, COBOL, FORTGI, FORTHX, GOFORT, and PLIOPT. Each allows the specification of a parameter list that may contain CMS options as well as compiler options that are identical to those coded on an OS EXEC card when invoking the compiler from OS.

Program execution is controlled by CMS commands such as RUN, INCLUDE, and LOAD. A core-image copy of the program can be recorded on disk with GENMOD, and later retrieved for execution with LOADMOD. Libraries to be searched during program compilation or load are specified by the GLOBAL command.

## CONTROL COMMANDS

The CMS user is able to define certain system parameters with the SET command. The parameters include: the amount of information to be included in the message printed at the end of command processing, the type of error messages to be printed at the terminal, and whether unknown commands should be passed on to CP. With the QUERY command, the user is given the current status of these and other CMS parameters. Synonyms for command names may be created by a user via entries in a CMS file with a filetype of SYNONYM. The EXEC command specifies a file of CP and CMS commands, as well as conditional branching and control statements, which will be executed in a predetermined sequence by the EXEC processor of CMS.

## LANGUAGE PROCESSORS

A VM/370 Assembler is distributed as a part of the VM/370 system and is required for installation and support. It is also used to assemble users' problem programs. All necessary macros for installation and support are provided in CMS libraries.

A variety of programming languages are available for use with CMS. BASIC, FORTRAN, and PL/I are useful languages for problem-solving applications, while COBOL, assembler language, and again PL/I are useful for commercial program development applications.

The compilers that can execute under CMS include OS PL/1, OS COBOL, VS BASIC, and OS FORTRAN IV. For information on programming languages and compilers that can be used with CMS, see "Appendix C: Language Processors and Emulators."

The compilers are invoked within the conversational environment of CMS; the normal mode of execution is to run the compilation to completion, type any diagnostic messages at the terminal, and make the listing file available for inspection at the terminal or for printing on the real printer.

Most object programs produced and compiled under CMS may be executed under CMS for direct problem solving. Programs that use certain OS system functions, described in the following paragraphs, must be run under the appropriate operating system.

To support the OS compilers, CMS simulates the execution of many of the OS macros. The sequential, direct, and partitioned access methods are logically simulated; the data records are physically maintained in the chained 800-byte blocks that are standard to CMS and are processed internally to simulate OS data set characteristics. Many OS Supervisor Call functions including GETMAIN/FREEMAIN and TIME are simulated.

The OS macros that are not simulated include those that support the Indexed Sequential Access Method and the telecommunications access methods. An OS problem program that uses only those functions for which simulation code exists may be tested and run under CMS. For example, all FORTRAN IV (G1) language functions will execute under CMS while COBOL programs that use ISAM will not.

Functions related to multi-tasking are either ignored by CMS or modified to achieve single task execution. See the _VM/370: System Programmer's Guide_ for more information about CMS's OS macro support.

## ALTERNATING OPERATING SYSTEMS

If a program to be tested uses OS functions that are not simulated, or if the program is designed for some other operating system (such as DOS), the user may execute the two operating systems alternately. The virtual machine must be configured to run both CMS and the other operating system.

Using this technique, the user first loads the Conversational Monitor System into the virtual machine. The Editor is used to make any necessary updates to the source program. Spooling facilities are used to copy the program (integrated into a suitable operating system job stream) into the virtual card reader. The user then issues the IPL command to load his other operating system (such as DOS) and begin the compilation. When the job stream has been processed, CMS is reloaded with the IPL command. The spooled printer output generated by the other operating system can be read onto a CMS user disk, inspected for diagnostic messages, then optionally scheduled for printing. Corrections and additional compilations, if necessary, follow the procedure above.

With this technique, a user can compile OS ISAM programs under CMS and then test them under OS. See the discussion "Dynamically Modified Channel Programs" in the _VM/370: Planning and System Generation Guide_.

## DEBUGGING FACILITIES

The debugging facilities of CMS permit a user to set instruction address stops in his program, to examine and modify virtual registers and virtual storage, and to trace all SVC interrupts. User-selected interrupts may be traced with output directed to either a virtual printer or the terminal.

Symbolic debugging capabilities are available to FORTRAN programmers using Code and Go FORTRAN or FORTRAN IV (G1) in the FORTRAN Interactive Debug Program Product, and to COBOL programmers using ANS Version 4 COBOL in the OS COBOL Interactive Debug Program Product.

## CMS BATCH FACILITY

The CMS Batch Facility is a VM/370 programming facility that runs under the CMS subsystem. It allows a VM/370 user to run jobs in batch mode by queuing jobs from either his own virtual machine or the real card reader to a virtual machine dedicated to running batch jobs under the Batch Facility. The Batch Facility virtual machine then executes these jobs, freeing the user's virtual machine for other uses. The accounting routines charge the time used in the batch machine to the originating user.

A Batch Facility virtual machine is generated and controlled at a terminal console under a userid dedicated to execution of jobs in batch mode. The system operator generates a batch machine by performing an IPL of CMS, then entering a command (CMSBATCH), which specifies that the machine is to execute jobs in batch mode. After each job is executed, the Batch Facility reloads itself, thereby providing a continuously running batch machine. Jobs are queued to the batch machine's virtual card reader from either user terminals or the system card reader and are executed sequentially. When the last job is executed, the Batch Facility waits for more input.

The Batch Facility is designed for the non-CMS user who requires a system for compiling or executing batch jobs loaded from the real system card reader. The Batch Facility is also useful for the interactive user who has compute-bound jobs such as assemblies and compilations, and for execution of large user programs. This allows interactive users to continue work at their terminals while their time-consuming jobs are run in another virtual machine.

Any user program written in a language supported by CMS can be executed on the batch machine. The same restrictions that apply to programs run under CMS, apply to programs run under the CMS Batch Facility. Also, there are restrictions on programs using certain CP and CMS commands. For full information about the CMS Batch Facility, see the <u>VM/370</u>: <u>Command Language Guide for General Users</u>.

# Remote Spooling Communications Subsystem

The VM/370 Remote Spooling Communications Subsystem (RSCS) is the VM/370 component that provides for spooling of files between remote stations and virtual machines at the VM/370 installation. (Remote stations are configurations of I/O devices attached to the VM/370 computer by binary synchronous (BSC) switched or nonswitched lines.)

The line drivers are components of RSCS designed to "drive," or control, a specific type of remote station.

Figure 13 shows the relationship between the VM/370 virtual machine users, the CP spool system, and the remote stations.

## The RSCS Virtual Machine

RSCS has a supervisor and also "line drivers." The supervisor, in conjunction with the CP spool system, provides for communication between the local VM/370 virtual machines and remote stations.

## The RSCS Teleprocessing Network

The RSCS network consists of the VM/370 computer, transmission control units attached to the central VM/370 computer, data sets and BSC telecommunications lines, and remote stations.

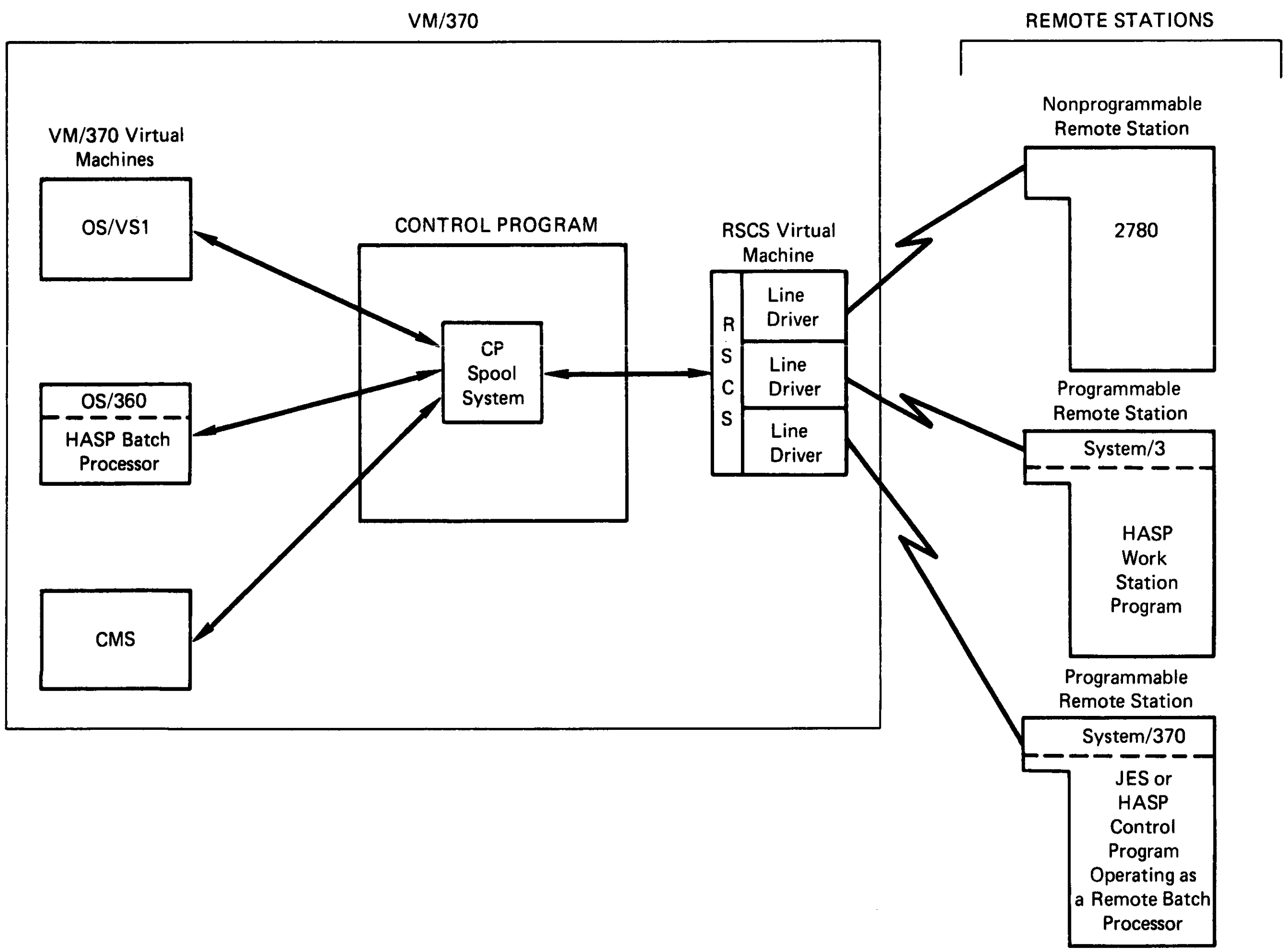


Figure 13. A VM/370 RSCS Teleprocessing Network

## THE VM/370 COMPUTER

The VM/370 computer is the functional center of communications in the RSCS teleprocessing network. The operator of the RSCS virtual machine controls the network by issuing RSCS commands at the RSCS virtual machine console.

The CP spool system is the logical center of the RSCS teleprocessing network. All files transmitted between remote locations and VM/370 virtual machines are routed through the CP spool system via the RSCS virtual machine.

## RSCS TELEPROCESSING HARDWARE REQUIREMENTS

To control the teleprocessing network, the VM/370 computer must have teleprocessing equipment (transmission control units, data sets, and communications lines) attached to it. Transmission control units control the transmission of data between the VM/370 computer and remote stations over communications lines. Data sets are devices that code and decode binary data for transmission over the communications lines. The specific devices supported by VM/370 for these functions are described in the *VM/370: Remote Spooling Communications Subsystem (RSCS) User's Guide*.

## REMOTE STATIONS

RSCS remote stations are I/O configurations. The minimum configuration consists of a card reader, a printer, and a card punch. There are two types of remote stations: programmable remote stations and nonprogrammable remote stations.

### Programmable Remote Stations

Programmable remote stations are I/O configurations that include a computer, such as a System/3, System/360, or System/370. If this computer is running a HASP-type or ASP-type batch processor, the remote station can receive files transmitted across the RSCS network, process the files, and transmit the results of the processing back to the originating location. Otherwise, the programmable remote station can only receive, read, print, punch, and send files. In other words, it acts as if it were nonprogrammable.

### Nonprogrammable Remote Stations

Nonprogrammable remote stations are I/O configurations that cannot be programmed, but can receive, read, print, punch, and send files. An example of a nonprogrammable remote station is a 2780 Data Transmission Terminal.

The types of devices supported for all types of remote stations, programmable and nonprogrammable, are listed in "Appendix B. System Requirements" and in the *VM/370: Remote Spooling Communications Subsystem (RSCS) User's Guide*.

# Using RSCS

The facilities of RSCS are selected and controlled by means of commands and control cards. Connections between geographically remote locations are made by the operator of the RSCS virtual machine.

## LINKING GEOGRAPHIC LOCATIONS IN THE RSCS NETWORK

Each location in the RSCS network is assigned a location identifier, which RSCS uses to find a link, or path, to the remote location.

In order to link a remote station to a virtual machine, the RSCS operator issues the START command for the requested linkid. Then, in order to begin transmitting to the RSCS, the remote station operator transmits a SIGNON card to the RSCS virtual machine.

Once the link between a remote station and the RSCS virtual machine is established via START and SIGNON, transmission to and from that location can begin. Each file transmitted from the remote station to the RSCS virtual machine is preceded by an ID card that specifies the destination of that file. Using the ID card, the file can be transmitted to any VM/370 virtual machine or, specifying the userid of the RSCS virtual machine and TAG information on the ID card, the file can be transmitted to another remote station in the RSCS network.

Figure 14 shows how the remote stations are linked to VM/370 virtual machines (including the RSCS virtual machine) and to other remote locations.

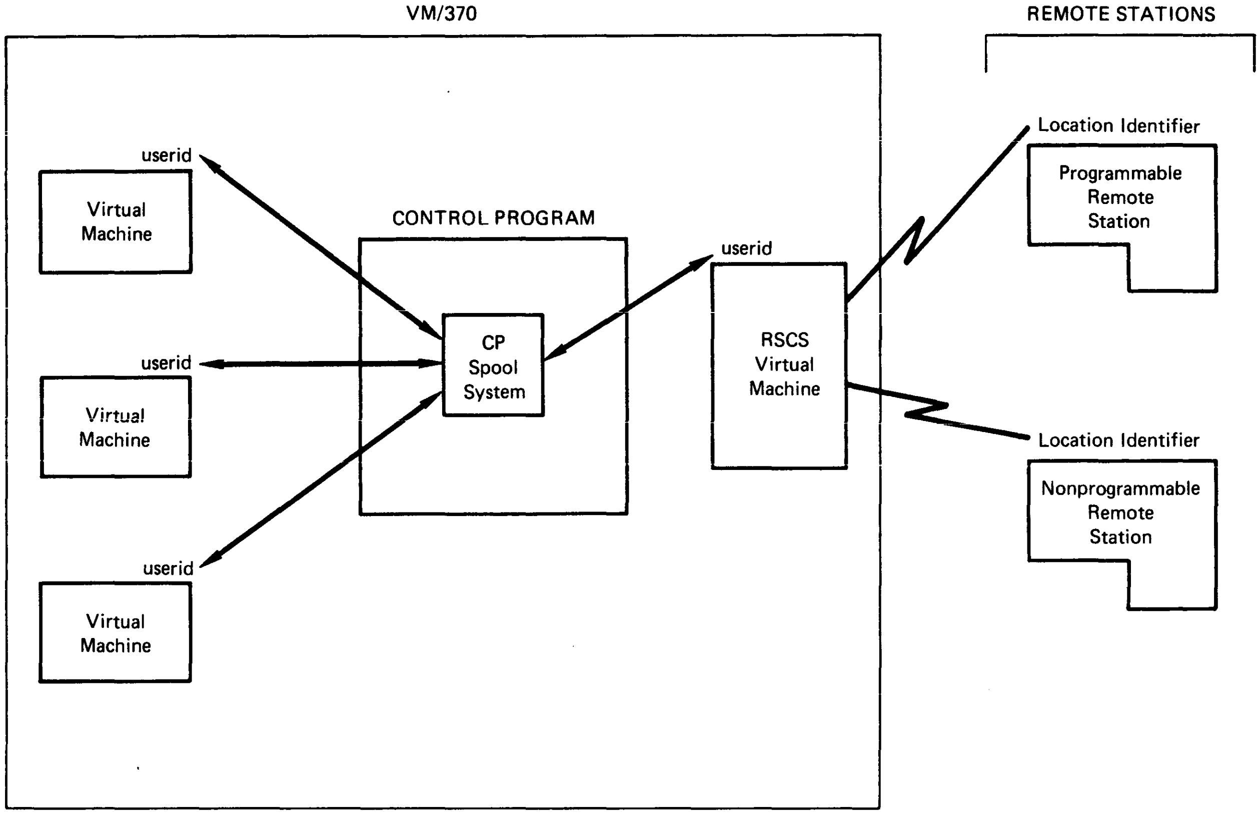


Figure 14. Linking Virtual Machines and Remote Stations

## COMMANDS USED TO TRANSMIT FILES AND TO CONTROL THE RSCS NETWORK

Transmission of files and system control are the two main functions of the RSCS Control Program.

### Commands for Transmitting Files

The CP TAG and SPOOL commands are used under RSCS to transmit files across a teleprocessing network. Virtual machine users use the CP TAG command to name the location identifier of the destination of the file to be transmitted. The CP SPOOL command and a command that closes the file to be transmitted, such as the CMS PUNCH or PRINT command, are used to enqueue the file on the RSCS virtual machine for processing and transmitting across the network. These commands perform the same function as the ID card used to transmit files from the remote locations to the virtual machine.

### Commands for Controlling the RSCS Virtual Machine and Remote Stations

RSCS provides commands and control cards for controlling the operation of the RSCS system. The system control functions are executed by the RSCS Control Program when it receives commands entered either from a console or via punched cards.

The RSCS virtual machine operator can use all the RSCS commands to control the system; operators at remote stations use a subset of the commands available to the RSCS virtual machine operator. In general, such functions as purging a file from the system, defining or deleting a link in the system, repositioning a file forward or backward during processing, disconnecting the RSCS virtual machine console, and so on, are provided by the system. These commands are described in detail in the *VM/370: Remote Spooling Communications Subsystem (RSCS) User's Guide*.

### Transmitting Commands To Be Executed by Other Systems

Batch systems with remote job entry capability, such as HASP and JES2, provide control commands unique to their systems. These commands are executed by means of commands and control cards entered at stations communicating with those batch systems. When RSCS is operating as a remote job entry system to one of the batch processors, the RSCS operator can issue commands to that batch processor by using the RSCS CMD command. The CMD command causes a HASP or ASP command to be transmitted to the batch processor for execution, just as if the command were transmitted to the processor by one of its work statons.

# Appendix A: Recovery Management Support

The IBM System/370 and its System Control Programs provide an extensive group of advanced reliability, availability, and serviceability (RAS) features. These features address the system requirements demanded by online and time-sharing operations.

The objective of the RAS features of the IBM System/370 is to reduce the frequency and impact of system interruptions caused by machine failure. In many cases, the system can be run in a degraded mode so that maintenance can be deferred to scheduled maintenance periods. When permanent failures do occur, their impact can be reduced by fast isolation and repair of the malfunction. RAS features are as follows:

1. Recovery facilities are provided to reduce the number of failures that cause a complete system termination. This permits deferred maintenance.

2. Repair procedures include online diagnosis and correction and/or circumvention of malfunctions concurrent with VM/370 execution. The effect of such repairs on system availability is thus reduced.

## System/370 Recovery Feature

The IBM System/370 attempts correction of most machine errors without program assistance. VM/370 is notified, via a machine check interruption, of both hard machine checks (permanent errors) and soft machine checks (transient errors). VM/370 then initiates error recording and error recovery procedures.

The following recovery features are implemented in the IBM System/370:

1. CPU retry of failing CPU operations.

2. Validity checking by Error Correction Codes on processor and control storage to correct all single-bit errors.

3. Input/output operation retry facilities including an extended channel status word (ECSW), which provides channel retry data to channel and control unit retry procedures.

4. Expanded machine check interrupt facilities to improve error recording and recovery procedures.

A detailed description of the recovery facilities for any particular IBM System/370 model can be obtained in the systems guide for that CPU, for example, the publication _A Guide to the IBM System/370 Model 145_, Order No. GC20-1734.

### AUTOMATIC RECOVERY FEATURES

The two hardware features that try to correct machine malfunctions before passing control to the Machine Check Handler are CPU Retry and Error Correction Codes (ECC). CPU Retry allows the machine to recover from transient CPU errors (soft machine checks) that would otherwise make it necessary to reload the operating system or terminate the executing program. ECC detects and often corrects invalid data fetched from processor and control storage.

#### _CPU Retry_

CPU errors are automatically retried by microprogram routines. These routines save source data before it is altered by the operation. When an error is detected, a microprogram returns the CPU to the beginning of the operation, or to a point where the operation was executing correctly, and the operation is repeated. After several unsuccessful retries, the error is considered permanent and causes a machine check interruption.

#### _Error Correction Codes (ECC)_

ECC checks the validity of data from processor and control storage, automatically correcting single-bit errors. It also detects multiple-bit errors but does not correct them.

As data enters and leaves storage, ECC checks each doubleword for correct parity in each byte. If a single-bit error is detected, ECC corrects it. The corrected doubleword is then sent back into processor or control storage and on to the CPU.

If a multiple-bit storage error is detected, a machine check interruption occurs, and information about the error is stored. The Machine Check Handler then gains control and determines, from the stored information, what action to take. (See the four possible actions listed in the section "Machine Check Handler" below.)

## VM/370 Recovery Features

The primary objectives of VM/370's Recovery Management Support are:

1. To reduce the number of system terminations that result from machine malfunctions.
2. To minimize the impact of such incidents.

These objectives are accomplished by programmed recovery which allows system operations to continue whenever possible, and by recording of the system status for all errors. The recovery management functions of VM/370 are provided by MCH (Machine Check Handler) and CCH (Channel Check Handler).

### MACHINE CHECK HANDLER (MCH)

A machine malfunction can originate from the CPU, processor storage, or control storage. When any of these fails to work properly, the hardware attempts to correct the malfunction. (See "Automatic Recovery Features" above.) If the machine recovers from the error through its own recovery facilities, the Machine Check Handler is notified by a machine check interruption. MCH records the fact that the machine has failed to operate properly. Concurrently with the machine check interruption, the CPU logs out fields of information in processor storage detailing the cause and nature of the error. MCH analyzes this information and builds the machine check record.

If the machine fails to recover from the error through its own recovery facilities, a machine check interruption occurs, and an interruption code indicates that the recovery attempt was unsuccessful. MCH then analyzes the data and attempts to keep the system as fully operational as possible. The cause of the malfunction determines what action MCH takes:

1. Resume operations leaving no adverse effects on the system.
2. Resume system operations at the expense of the user that was interrupted.
3. Isolate the failure to a page and flag the page as invalid or unavailable for use by the Paging Supervisor.
4. Automatic restart with spool files saved or place the system in a disabled wait state.

### CHANNEL CHECK HANDLER (CCH)

The Channel Check Handler is a resident program that receives control from the I/O supervisor when a real channel error occurs. CCH records the error. Channel data checks associated with a virtual machine input/output operation are passed on to the virtual machine for handling; other types of channel errors cause termination of the virtual machine. Channel errors associated with an input/output operation initiated by CP (for example, paging or spooling) are retried by the appropriate device-dependent error recovery procedure if CCH determines that system integrity has not been damaged.

If CCH determines that system integrity has been damaged (for example, if the channel has been reset, or if the device address stored is invalid), the system is placed in a disabled wait state and a message is sent to the system resource operator.

### HANDLING OF HARD MACHINE CHECKS

If a permanent error (hard machine check) occurs, the error is analyzed to determine whether or not it is correctable by programming. Time-of-day clock and CPU timer errors that result in a machine check interruption are not correctable. A time-of-day clock error causes the real computing system to be placed in a disabled wait state. A CPU timer error causes VM/370 to be terminated, and then reinitialized by an automatic restart.

Uncorrectable or unretryable CPU errors, storage errors, and storage protect key failures are handled as discussed in the following paragraphs.

### CPU Errors

When a machine check interruption indicates that an uncorrectable or unretryable CPU error associated with VM/370 has occurred, the system operator is informed of the error and the system is terminated. An automatic restart reinitializes VM/370. All virtual machine users must log on again. If the error is associated with a virtual machine, the user is informed of the error and the virtual machine is reset, unless it is using the virtual=real option. In that case, the virtual machine is terminated, and the user must then log on and re-IPL his machine.

### Storage Errors in a Virtual Machine Page

When the control program detects a permanent error (hard machine check) in a real storage page frame that is being used by a virtual machine, the frame is marked invalid if the error is intermittent, or unavailable if the error is solid. If the page frame has not been altered by the virtual machine, a new page frame is assigned to the virtual machine and a backup copy of the page is brought in the next time the page is referenced. The storage error is transparent to the virtual machine user.

If the page frame has been altered, VM/370 resets the virtual machine, clears its virtual storage to zeroes, and sends an appropriate message to the user. If the virtual machine is using the virtual=real option, it is terminated. In either case, normal system operation continues for all other users.

### Storage Errors in the CP Nucleus

Multiple-bit storage errors in the CP nucleus cannot be refreshed; they cause VM/370 to be terminated. An automatic restart reinitializes VM/370. (Single-bit storage errors are corrected by ECC, as noted above.)

### Storage Protect Key Failure

When intermittent storage protect key failures occur, whether associated with VM/370 or a virtual machine, the key is refreshed and operation continues.

If the storage protect key failure is uncorrectable (solid) and is associated with a virtual machine, the user is notified and the virtual machine is terminated. The page frame is marked unavailable. Uncorrectable (solid) storage protect key failures associated with VM/370 cause the VM/370 system to be terminated. An automatic restart reinitializes VM/370.

## HANDLING OF SOFT MACHINE CHECKS

Although hard machine checks always cause a machine check interruption to occur and logouts to be written, soft machine checks are handled in one of two ways depending on which mode of operation the VM/370 system is in.

The two possible modes of system operation are recording mode and quiet mode. In recording mode, soft machine checks cause machine check interruptions to occur and logouts to be written. In quiet mode, only hard machine checks cause machine check interruptions and logouts.

The initialized state (and therefore, the normal operating state) of VM/370 for CPU Retry reporting is recording mode. For ECC reporting, the initialized state of VM/370 is model dependent: quiet mode for Models 135, 145, 158, and 168, and recording mode for Models 155 II and 165 II.

A change from RECORD to QUIET mode can occur in one of two ways: when 12 soft machine checks have occurred, or when the SET MODE RETRY/MAIN QUIET command is executed by maintenance personnel. A change from QUIET to RECORD mode occurs when the SET MODE RETRY/MAIN RECORD command is executed by maintenance personnel.

If a transient error (soft machine check) occurs while the system is in RECORD mode, a machine check record containing information about the error is written on the MCH/CCH error cylinder. This record includes the data in the fixed logout area, the date, the time of day and other pertinent data. The operator is not informed that a soft machine check has occurred.

If a transient error occurs while the system is in QUIET mode, no machine check interruption occurs, and no logouts are written. The hardware, which had gained control when the soft machine check occurred, returns control to either VM/370 or the problem program, whichever had control at the time the machine check occurred.

## ERROR RECOVERY PROCEDURES

Device-dependent error recovery procedures are included in VM/370 for all devices supported by VM/370. Functionally, these procedures perform as do their counterparts in an OS or DOS system. The standards used by OS or DOS for priority of error testing, recommended retry action, and number of retry attempts for a particular error type are used by VM/370. The error recovery procedures accept and use the extended channel status word, determine if retry is possible, and initiate retry actions.

### CP Input/Output Errors

An appropriate error recovery procedure is invoked whenever an error occurs which is related to a CP input/output operation, such as paging or spooling. If the error cannot be corrected, it is recorded, and the system resource operator is notified of the error.

### Handling of Virtual Machine Input/Output Errors

channel data checks, and interface control checks) associated with virtual machine START I/O requests are passed to the virtual machine. The machine operating system assesses the error and attempts retry.

Note that CMS uses the Diagnose interface to request VM/370 to perform input/output operations, and VM/370 then performs any necessary recovery operations for errors associated with the request.

### Recording Virtual Machine Input/Output Errors

By use of the SVC 76 error recording interface, VM/370 provides uniform recording of errors encountered by operating systems running in virtual machines. VM/370 records the real address (rather than the virtual address) of a device that experiences an error, so that it can be located by support personnel. The operating systems that use the SVC 76 interface are:

VM/370 Release 2 and above
  (running in a virtual machine)
DOS/VS
OS/VS1
OS/VS2
OS Release 21.0 and above
DOS Release 27.0 (requires PTF #1124)
DOS Release 27.1 (requires PTF #2051)

When an SVC 76 is issued, CP examines the error record built by the virtual machine operating system. If the information is valid, CP translates from virtual to real device addresses and then records the error information on the VM/370 error recording cylinder. If this information is invalid, the SVC is reflected back to the virtual machine and no recording takes place. Duplicate recording of errors is thus avoided.

In case of a permanent I/O error, VM/370 sends a message to the primary system operator.

If a virtual machine is using one of the above operating systems and is also using the Virtual Machine Assist feature, then all SVCs are handled by the Assist feature (except SVC 76, which is always handled by CP). However, the user can specify that CP handle all SVCs by issuing SET ASSIST NOSVC, or by including the SVCOFF option in his directory entry.

If a virtual machine is using an operating system that does not use the SVC 76 interface, both CP and the virtual machine record errors.

For more informaticn about the SVC 76 error recording interface and the Virtual Machine Assist feature, refer to the *VM/370: System Programmer's Guide*.

## RECORDING FACILITIES

The CP Environmental Recording, Editing, and Printing program (CPEREP) is included as part of the VM/370 system and is run as a problem program under CMS. The printed output from CPEREP under VM/370 has the same format as that generated by OS, OS/VS1, and OS/VS2. The system can:

- Edit and print all cr specified error records contained on the system error recording file.

- Create a history of records on an accumulation tape that is used as input to analysis programs that run under OLTSEP.
- Erase all or specified areas of the error recording file.

## VM/370 Repair Facilities

The Online Test Standalone Executive Program (OLTSEP) and Online Tests (OLT) execute in a virtual machine that runs concurrently with normal system operations. These programs provide online diagnosis of input/output errors for most devices that attach to the IBM System/370.

The service representative can execute online tests from a terminal as a user of the system; VM/370 console functions, including the ability to display or alter the virtual machine storage, are available when these tests are run. Those tests that violate VM/370 restrictions may not run correctly in a virtual machine environment.

## VM/370 Restart Facilities

When either MCH or CCH determines that an error has damaged the integrity of VM/370, the system IPL routine is called to perform an orderly termination of the system and subsequent reloading and initialization of VM/370. The system attempts to execute a warm start, thus allowing terminal lines to be re-enabled automatically and completed spool files to be maintained. Storage reconfiguration data (such as page frames marked unavailable or invalid), which is acquired during the process of recovering from real storage errors, is lost. After a VM/370 system failure, each user must reinitialize his virtual machine.

Resetting of a virtual machine, whether caused by a real computing system malfunction or by a virtual machine program error, does not affect the execution of other virtual machines, unless they are sharing the area in which the malfunction occurred.

# Appendix B: System Requirements

## Machine Requirements

The following machines and devices are supported by VM/370. Specific information regarding model numbers and supported features can be provided by an IBM marketing representative.

CPUS

IBM System/370 Model 135
IBM System/370 Model 145
IBM System/370 Model 155 II
IBM System/370 Model 158
IBM System/370 Model 165 II
IBM System/370 Model 168

These machines must be equipped with the Dynamic Address Translation facility, the System Timing facilities, and at least 245,760 bytes of real storage. They must operate in extended control mode. The Models 135 and 145 require the floating-point feature. The Models 165 II and 168 require the Channel Indirect Data Addressing feature on each of the following standalone channels: 2860, 2870, and 2880.

The resident portion of the VM/370 control program requires approximately 112K bytes of real storage plus an average of 2.8K bytes per active virtual machine.

For more information about storage requirements, refer to the _VM/370: Planning and System Generation Guide_.

SYSTEM CONSOLES

The following system consoles as well as the terminals listed below are supported by VM/370 as virtual consoles (simulated as 3215 consoles).

IBM 2150 Console with 1052 Printer-Keyboard, Model 7
IBM 3066 System Console for the System/370 Models 165 II and 168
IBM 3210 Console Printer-Keyboard, Models 1 and 2
IBM 3215 Console Printer-Keyboard, Model 1
IBM 7412 Console (via RPQ AA2846) with a 3215, Model 1
IBM System Console for the System/370 Model 158 (in printer-keyboard mode with the 3213 Printer Model 1 required, or in display mode)

TELECOMMUNICATIONS

_Terminals_

The following devices are supported for use as virtual system consoles (and, consequently, as CMS user terminals):

IBM 1050 Data Communication System
IBM 2741 Communication Terminal
| IBM 3277 Display Station, Model 2 (Local and Remote Attachment)
Line Control for CPT-TWX (Model 33/35) terminals
IBM 3767 Communication Terminal, Models 1 and 2 (2741 compatible-equipped)

The following device is supported for remote spooling and the entry of spool commands:

IBM 2780 Data Transmission Terminal Model 2

(See also "System Support for the RSCS Teleprocessing Network" below.)

_Terminal Control Unit_

| IBM 3271 Control Unit, Model 2 (Remote Attachment)
IBM 3272 Control Unit, Model 2 (Local Attachment)

_Transmission Control Units_

IBM 2701 Data Adapter Unit
IBM 2702 Transmission Control
IBM 2703 Transmission Control
IBM 3704/3705 Communications Controllers (In Network Control Program mode, Partitioned Emulation Program mode, and 2701, 2702, 2703 Emulation Program mode)
IBM Integrated Communications Adapter (#4640) available on the System/370 Model 135

## System Support for the RSCS Teleprocessing Network

VM/370 provides device and subsystem support for both programmable and nonprogrammable RSCS remote stations.

- Devices Supported for Nonprogrammable Remote Stations

  IBM 2770 Data Communication System, with the 2772 Multipurpose Control Unit
  IBM 2780 Data Transmission Terminal, Models 1 and 2
  IBM 3770 General Purpose Communication Terminal, Model 2 (operating in 2770 compatibility mode)
  IBM 3780 Data Communications Terminal

- Transmission Control Units Supported for Programmable Remote Stations

  IBM 2701 Data Adapter Unit with Synchronous Data Adapter, Type II
  IBM 2703 Transmission Control Unit with Synchronous Terminal Control
  IBM 3704 Communications Controller (in EP mode only)
  IBM 3705 Communications Controller (in EP mode only)

- Systems Supported for Remote Job Entry into RSCS

  IBM System/360 Models 20, 22, 25, 30, 40, 50, 65, 75, 85, 195
  IBM System 370 Models 115, 125, 135, 145, 155, 155 II, 158, 165, 165 II, and 168
  IBM 1130 System
  IBM System 3 Model 10
  IBM 2922 Programmable Terminal

When RSCS is operating as a remote job entry system to a HASP-type or ASP-type batch processor, it supports any IBM system that supports the HASP/ASP-type system.

- Software Subsystems Supported for RSCS

  HASP II Version 3.1 (360D-05.1.014)
  HASP II Version 4 (370H-TX-001)
  ASP Version 3.1 (360A-CX-15X)
  JES2 Component of VS2 Release 2
  JES3 Component of VS2 (when available)
  RES Component of VS1 Release 2 and above

DIRECT ACCESS STORAGE DEVICES

IBM 2305 Fixed Head Storage, Model 1 (Models 165 II and 168 only) and Model 2
IBM 2314 Direct Access Storage Facility
IBM 2319 Disk Storage
IBM 3330 Disk Storage, Models 1, 2, and 11
IBM 3333 Disk Storage and Control, Models 1 and 11
IBM 3340 Direct Access Storage Facility, Models A2, B1, and B2, with Data Modules Models 35, 70, and 70F

DIRECT ACCESS CONTROL UNITS

IBM 2835 Storage Control Model 1 for 2305 Model 1 (Models 165 II and 168 only)
IBM 2835 Storage Control Model 2 for 2305 Model 2
IBM 2844 Auxiliary Storage Control for 2314 and 2319
IBM 3333 Disk Storage and Control Models 1 and 11 for the 3330 Models 1, 2, and 11
IBM 3345 Integrated Storage Control Models 3, 4, and 5 on the Model 145 for 3330 Disk Storage Models 1 and 2, and 3333 Disk Storage and Control, Models 1 and 11
IBM 3830 Storage Control Model 1 for 3330 Models 1 and 2 only
IBM 3830 Storage Control Model 2 for 3333 Models 1 and 11
IBM IFA (Integrated File Adapter) (#4650) on System/370 Models 135 and 145 for 2319
IBM IFA (Integrated File Adapter) (#4655) on the Model 135 for 3330 Models 1 and 2, 3333 Models 1 and 11, and 3340 Model A2
IBM ISC (Integrated Storage Control) on the Model 158 for 3330 Models 1 and 2, 3333 Models 1 and 11, and 3340 Model A2
IBM ISC (Integrated Storage Control) on the Model 168 for 3330 Models 1 and 2, 3333 Models 1 and 11, and 3340 Model A2

All direct access devices are supported as VM/370 system residence, paging, and spooling devices. All except the 2305 are supported by CMS.

MAGNETIC TAPES

IBM 2401, 2402, 2403 Magnetic Tape Units
IBM 2415 Magnetic Tape Unit, Models 1, 2, 3, 4, 5, and 6
IBM 2420 Magnetic Tape Unit, Models 5 and 7
IBM 3410 Magnetic Tape Units, Models 1, 2, and 3
IBM 3420 Magnetic Tape Unit, Models 3, 4, 5, 6, 7, and 8

MAGNETIC TAPE CONTROL UNITS

IBM 2803 Tape Control
IBM 2804 Tape Control
IBM 3411 Magnetic Tape Unit and Control, Models 1, 2, and 3
IBM 3803 Tape Control

PRINTERS

IBM 1403 Printer, Models 2, 3, 7 and N1
IBM 1443 Printer, Model N1
IBM 3211 Printer
IBM 3284 Printer, Models 2 and 3 (remote 3270)
IBM 3286 Printer, Models 2 and 3 (remote 3270)
IBM 3288 Line Printer, Model 2 (remote 3270)

READER/PUNCHES

IBM 2501 Card Reader, Models B1 and B2
IBM 2520 Card Punch, Models B2 and B3
IBM 2540 Card Read Punch, Model 1
IBM 3505 Card Reader, Models B1 and B2
IBM 3525 Card Punch, Models P1, P2 and P3

UNIT RECORD CONTROL UNITS

IBM 2821 Control Unit
IBM 3811 Printer Control Unit
IBM IPA (Integrated Printer Adapter) for the 1403 Printer on the Model 135

MINIMUM VM/370 CONFIGURATION

| | |
|---|---|
| CPU | One of the System/370 Models designated |
| Storage | 245,760 bytes |
| One | Console device |
| One | Printer |
| One | Card Reader |
| One | Card Punch |
| Two | Spindles Direct Access Storage |
| One | Nine-track Magnetic Tape Unit |
| One | Telecommunications Control Unit (or the Integrated Communications Adapter on the System/370 Model 135) or one 3272 Control Unit (if only 3277 Display Stations are used) |
| One | Multiplexer Channel |
| One | Selector or Block Multiplexer Channel |
| One | Communications Terminal |

## VM/370 Programming Characteristics

VM/370 is written in VM/370 Assembler language, using the instructions available only on an IBM System/370 with dynamic address translation. CMS command modules are written in VM/370 Assembler language.

## Support Considerations

CMS is used to incorporate all VM/370 program releases and updates.

## VM/370 Assembler

A VM/370 Assembler is distributed as a part of the VM/370 system and is required for installation and further support of the system. All necessary installation and support macros are provided in CMS libraries.

The Conversational Monitor System (CMS) is a component of VM/370 and is distributed with it. Certain other facilities mentioned in this publication are not part of VM/370, but can be separately ordered from IBM. These include: IBM System/360 and System/370 operating systems, IBM language processors and other program products, IBM Installed User Programs, and IBM Field Developed Programs. For more information, contact your IBM representative.

## Program Products

Figure 15 lists the IBM program products that are executable under CMS.

To find the amount of storage required to install a program product, refer to the appropriate program product publication.

## Installed User Programs

A text-processing program is available: IBM Installed User Program (IUP) SCRIPT/370 (IBM Program No. 5976-PAF). SCRIPT/370 creates formatted output from one or more CMS files, each of which contains text and/or SCRIPT control words. The SCRIPT files are created and modified at a terminal using the CMS Editor.

SCRIPT/370 manuscript facilities include right margin justification, line centering, inserting top and bottom titles, and the ability to invoke additional SCRIPT input files from the file being processed. Other facilities to assist in the preparation of large documents include symbolic capabilities that can generate a table of contents, and number pages and figures.

| IBM Program Product | IBM Program Number |
|---|---|
| OS Code & Go FORTRAN | 5734-FO1 |
| OS FORTRAN IV (G1) | 5734-FO2 |
| OS FORTRAN IV Library (Mod I) | 5734-LM1 |
| OS FORTRAN IV (H) Extended | 5734-FO3 |
| OS FORTRAN Library (Mod II) | 5734-LM3 |
| FORTRAN Interactive Debug | 5734-FO5 |
| OS/VS COBOL Compiler and Library | 5740-CB1 |
| OS/VS COBOL Library Only | 5740-LM1 |
| OS Full American National Standard COBOL Version 4 Compiler and Library | 5734-CB2 |
| OS Full American National Standard COBOL Version 4 Library | 5734-LM2 |
| OS COBOL Interactive Debug | 5734-CB4 |
| OS PL/I Optimizing Compiler | 5734-PL1 |
| VS BASIC Processor | 5748-XX1 |
| VS BASIC | 5734-XX1 |
| OS PL/I Resident Library | 5734-LM4 |
| OS PL/I Transient Library | 5734-LM5 |
| OS PL/I Optimizing Compiler and Libraries | 5734-PL3 |
| OS PL/I Checkout Compiler | 5734-PL2 |
| Planning Systems Generator/CMS (PSG/CMS) | 5748-XT1 |

Figure 15. IBM Program Products

A timesharing facility is available as an IUP: the McGill University System for Interactive Computing (MUSIC), IBM Program No. 5796-AAT. MUSIC is a conversational time-sharing operating system that can run in a virtual machine under VM/370.

| System/370 Model | 1401 1440 | 1401 1440 1460 | 1460 1410 7010 | 7070 7074 | 7080 | 709 7090 7094 7094II |
|---|---|---|---|---|---|---|
| 135 | #4457 | | | | | |
| 145 | #4457 | | #4458 | | | |
| 155 II,158 | | | #3950 | #7117 | | |
| 165 II,168 | | | | #7117 | #7118 | #7119 |

Figure 16. Integrated Emulators that Execute Under VM/370

## Integrated Emulators

Emulator-dependent programs (except for DOS emulation under OS or OS/VS) that execute on a particular System/370 equipped with the appropriate compatibility features can execute on that System/370 in DOS or OS virtual machines under VM/370.

Figure 16 shows, by System/370 model number, which integrated emulators can execute under VM/370 and the compatibility feature numbers (#xxxx) that are required.

No changes are required to the emulators, to DOS or OS, or to VM/370 itself to allow emulator-dependent programs to execute in virtual machines.

On the System/370 Model 158 only, the Virtual Machine Assist Feature cannot operate concurrently with the 7070/7074 compatibility feature (Feature #7117).

# Appendix D: VM/370 Restrictions

A virtual machine created by VM/370 is capable of running an IBM System/360 or System/370 operating system as long as certain VM/370 restrictions are not violated. Virtual machine restrictions and certain execution characteristics are stated in this appendix.

## Dynamically Modified Channel Programs

In general, virtual machines may not execute channel programs that are dynamically modified (that is, channel programs that are changed between the time the START I/O (SIO) is issued and the end of the input/output occurs, either by the channel program itself or by the CPU). However, some dynamically modified channel programs are given special handling by CP: specifically, those generated by the Indexed Sequential Access Method (ISAM) running under OS/PCP, OS/MFT, and OS/MVT; those generated by ISAM running in an OS/VS virtual=real partition; and those generated by the OS/VS Telecommunications Access Method (TCAM) Level 5, with the VM/370 option.

The self-modifying channel programs that ISAM generates for some of its operations receive special handling if VM/370 is generated with the ISAM option and if the virtual machine using ISAM has that option specified in its VM/370 directory entry. There is no such restriction for DOS ISAM, or for ISAM if it is running in an OS/VS virtual=virtual partition. If ISAM is to run in an OS/VS virtual=real partition, you must specify the ISAM option in the VM/370 directory entry for the OS/VS virtual machine.

Virtual machines using OS/VS TCAM (Level 5, generated or invoked with the VM/370 option) issue a DIAGNOSE instruction when the channel program is modified. This instruction causes CP to reflect the change in the virtual CCW string to the real CCW string being executed by the channel. CP is then able to execute the dynamically modified channel program properly.

The restriction against dynamically modified channel programs does not apply if the virtual machine has the virtual=real performance option and the NOTRANS option has been set on.

## Minidisk Restrictions

The following restrictions exist for minidisks:

1. In the case of Read Home Address with the skip bit off, VM/370 modifies the home address data in user storage at the completion of the channel program because the addresses must be converted for minidisks; therefore, the data buffer area may not be dynamically modified during the input/output operation.

2. On a minidisk, if a CCW string uses multitrack search on input/output operations, subsequent operations to that disk must have preceding seeks or continue to use multitrack operations. There is no restriction for dedicated disks.

3. OS/PCP, MFT, and MVT ISAM may be used with a minidisk only if the minidisk is located at the beginning of the physical disk (that is, at cylinder zero). There is no such restriction for DOS or OS/VS ISAM.

4. VM/370 does not return an end of cylinder condition to a virtual machine that has a virtual 2311 mapped to the top half (that is, tracks 0 through 9) of 2314 or 2319 cylinders.

5. If the user's channel program (CCWs) for a minidisk do not perform a Seek operation, then to prevent accidental accessing, VM/370 inserts a positioning Seek operation into the user's CCWs. Thus, certain channel programs may generate a condition code (CC) of zero on a SIO instead of an expected CC of one, which is reflected to the virtual machine. The final status is reflected to the virtual machine as an interrupt.

6. DASD channel programs directed to minidisks on 3330 or 3340 devices may give different results than on dedicated drives if the channel program includes multiple-track operations and depends on a Search ID High or a Search ID High or Equal to terminate the program. This is because the record 0 count fields on

the 3330 and 3340 must contain the real cylinder number of the track on which they reside; therefore, a Search ID High based on a low virtual cylinder number may terminate prematurely if a real record 0 is encountered. This restriction does not apply to minidisks with a relocation factor of zero. This restriction does apply to minidisks with a VTOC greater than one track that are used with OS (Release 20.6 and later) or OS/VS (any release), since the VTOC Locate function uses a Search ID High to stop at the end of the VTOC.

<u>Note</u>: If the 'R' byte of 'CCHHR' is equal to zero at the time a virtual Start I/O is issued, but the 'CCHHR' field is read in dynamically by the channel program before the SEARCH ID CCW is executed, then the real SEARCH ID CCW uses the relocated 'CCHHR' field instead of the 'CCHHR' field that was dynamically read in. This causes erroneous results. To avoid this problem, the virtual machine should not default the'R' byte of 'CCHHR' to binary zero if the search arguments are to be read in dynamically and a SEARCH ID on Record R0 is not intended.

7. The IBCDASDI program cannot assign alternate tracks for a 3330.

## Timing Dependencies

Timing dependencies in input/output devices or programming do not function consistently under VM/370:

1. The following telecommunication access methods (or the designated option) violate the restriction on timing dependency by using program-controlled interrupt techniques and/or the restriction on dynamically modified channel programs:

   - OS Basic Telecommunications Access Method (BTAM) with the dynamic buffering option.
   - OS Queued Telecommunications Access Method (QTAM).
   - DOS Queued Telecommunications Access Method (QTAM).
   - OS Telecommunications Access Method (TCAM).
   - OS/VS Telecommunications Access Method (TCAM) Level 4 or earlier, and Level 5 if TCAM is not generated or invoked with the VM/370 option.

   These access methods may run in a virtual=real machine with CCW translation suppressed by SET NOTRANS ON. (OS BTAM can be generated without dynamic buffering, in which case no virtual machine execution violations occur.) However, the BTAM reset poll macro will not execute under VM/370 if issued from third level storage. For example, a reset poll macro has a NOP effect if executed from a virtual=virtual storage under VS1 which is running under VM/370.

2. Programming that makes use of the PCI channel interrupt for channel program modification or processor signalling must be written so that processing can continue normally if the PCI is not recognized until I/O completion or if the modifications performed are not executed by the channel.

3. Devices that expect a response to an interrupt within a fixed period of time may not function correctly because of execution delays caused by normal VM/370 system processing. An example of such a device is the IBM 1419 Magnetic Character Reader.

4. The operation of a virtual block multiplexer channel is timing dependent. For this reason, the channel appears available to the virtual machine operating system, and channel available interrupts are not observed. However, operations on virtual block-multiplexing devices should use the available features like Rotational Position Sensing to enhance utilization of the real channels. If a virtual machine has a dedicated block multiplexer channel, then that channel operates in true block-multiplexing mode for the virtual machine.

## CPU Model-Dependent Functions

On the System/370 Model 158 only, the Virtual Machine Assist feature cannot operate concurrently with the 7070/7074 compatibility feature (Feature #7117).

Programs written for CPU model-dependent functions may not execute properly in the virtual machine under VM/370. The following points should be noted:

1. Programs written to examine the machine logout area do not have meaningful data since VM/370 does not reflect the machine logout data to a virtual machine.

2. Programs written to obtain CPU identification (via the Store CPU ID instruction, STIDP) receive the real machine value. When the STIDP instruction is issued by a virtual machine, the version code contains the value 256 in hexadecimal ("FF") to represent a virtual machine.

3. Programs written to obtain channel identification (via the Store Channel ID instruction, STIDC) receive information from the virtual channel block. For dedicated channels, the real channel model and type and the length of the I/O extended logout area are reflected. For non-dedicated channels, however, only the virtual channel type is reflected; the other fields contain zeroes.

4. No simulation of other CPU models is attempted by VM/370.

## Virtual Machine Characteristics

Other characteristics that exist for a virtual machine under VM/370 are as follows:

1. If the virtual=real option is selected for a virtual machine, input/output operations specifying data transfer into or out of the virtual machine's page zero, or into or out of storage locations whose addresses are greater than the storage allocated by the virtual=real option, must not occur. The storage-protect-key mechanism of the IBM System/370 CPU and channels operates in these situations but is unable to provide predictable protection to other virtual machines. In addition, violation of this restriction may compromise the integrity of the system. The results are unpredictable.

2. VM/370 has no multiple path support and, hence, does not take advantage of the two-channel switch. However, a two-channel switch can be used between the IBM System/370 running a virtual machine under VM/370 and another CPU.

3. The DIAGNOSE instruction cannot be issued by the virtual machine for its normal function. VM/370 uses this instruction to allow the virtual machine to communicate system services requests. The Diagnose interface requires the operand storage addresses passed to it to be real to the virtual machine issuing the DIAGNOSE instruction. For more information about the DIAGNOSE instruction in a virtual machine, see the <u>VM/370: System Programmer's Guide</u>.

4. A control unit never appears busy to a virtual machine unless the unit is on a channel dedicated to the virtual machine.

5. The number of pages used for input/output must not exceed the total number of user pages available in real storage; violation of this restriction causes the real computing system to be put into an enabled wait state.

6. The CP IPL command cannot simulate self-modifying IPL sequences off dedicated unit record devices or certain self-modifying IPL sequences off tape devices.

7. The VM/370 spooling facilities do not support punch-feed-read, stacker selection, or column binary operations. Detection of carriage control channels is supported for a virtual 3211 only.

8. VM/370 does not support count control on the virtual 1052 operator's console.

9. Programs that use the integrated emulators function only if the real computing system has the appropriate compatibility feature. VM/370 does not attempt simulation. The DOS emulators are not supported.

10. The READ DIRECT and WRITE DIRECT instructions are not supported for a virtual machine.

11. The System/370 SET CLOCK instruction cannot be simulated and, hence, is ignored if issued by a virtual machine. The System/370 STORE CLOCK instruction is a nonprivileged instruction and cannot be trapped by VM/370; it provides the true TOD clock value from the real CPU.

12. The 1050/1052 Model 2 Data Communication System is supported only as a keyboard operator's console. Card reading, paper tape I/O, and other modes of operation are not recognized as unique, and hence may not work properly. This restriction applies only when the 1050 system is

used as a virtual machine operator's console. It does not apply when the 1050 system is attached to a virtual machine via a virtual 2701, 2702, or 2703 line.

13. The pseudo-timer (usually device address 0FF, device type TIMER) does not return an interrupt from a Start I/O; therefore, do not use EXCP to read this device.

14. A virtual machine IPL with the NOCLEAR option renders one page of the virtual machine invalid. The IPL simulator uses one page of the virtual machine to initiate the IPL function. The starting address of the invalid page is either the result of the following formula:

$$\frac{\text{virtual machine size}}{2} = \left\{\begin{array}{l}\text{starting}\\ \text{address of}\\ \text{invalid page}\end{array}\right.$$

or the hexadecimal value 20,000, whichever is smaller.

15. To maintain system integrity, data transfer sequences to and from a virtual system console are limited to a maximum of 2032 bytes. Channel programs containing data transfer sequences that violate this restriction are terminated with an interrupt whose CSW status indicates incorrect length and a channel program check.

    Note: A data transfer sequence is defined as one or more read or write CCWs connected via chain data. The introduction of command chaining defines the start of a new data transfer sequence.

16. If you intend to define more than 73 virtual devices for a single virtual machine, be aware that any single request for free storage in excess of 512 doublewords (a full page) will cause the VM/370 system to abnormally terminate (ABEND code PTR007) if the extra storage is not available on a contiguous page. Therefore, two contiguous pages of free storage must be available in order to log on a virtual machine with more than 73 virtual devices (three contiguous pages for a virtual machine with more than 146 virtual devices, etc.). Contiguous pages of free storage are sure to be available only immediately after IPL, before other virtual machines have logged on. Therefore, a virtual machine with more than 73 devices should be the first to log on after IPL.

17. I/O interrupts for a virtual channel-to-channel adapter that has both ends connected to the same virtual machine cannot be traced via the CP TRACE command.

18. When an I/O error occurs on a device, the System/370 hardware maintains a contingent connection for that device until a SENSE channel command is executed and sense data is recorded. That is, no other I/O activity can occur on the device during this time. Under VM/370, the contingent connection is maintained until the SENSE command is executed, but I/O activity from other virtual machines can begin on the device while the sense data is being reflected to the virtual machine. Therefore, the user should be aware that on a shared disk, the access mechanism may have moved during this time.

19. The mode setting for 7-track tape devices is maintained by the control unit. Therefore, when a virtual machine issues the SET MODE channel command to a 7-track tape device, it changes the mode setting of all 7-track tape devices attached to that control unit.

    This has no effect on virtual machines (such as OS or DOS) that issue SET MODE each time a CCW string is to be executed. However, it can cause a problem if a virtual machine fails to issue a SET mode with each CCW string executed. Another virtual machine may change the mode setting for another device on the same control unit, thereby changing the mode setting of all 7-track tape devices attached to that control unit.

20. OS/VS2 is supported in uniprocessor mode only.

## CMS Restrictions

The following restrictions apply to CMS, the conversational subsystem of VM/370:

1. CMS executes only on a virtual IBM System/370 provided by VM/370.

2. The maximum sizes of CMS minidisks are as follows:

| Disk | Maximum Cylinders |
|---|---|
| 2314/2319 | 203 |
| 3330 Series | 246 |
| 3340 Model 35 | 349 |
| 3340 Model 70 | 682 |

3. Unit record equipment cannot be dedicated to CMS; the spooling facilities of VM/370 must be used.

4. Only those OS facilities that are simulated by CMS can be used to execute OS programs produced by language processors under CMS.

5. Many types of object programs produced by CMS (and OS) languages can be executed under CMS using CMS's simulation of OS supervisory functions. The following functions, although supported in DOS and OS virtual machines under VM/370, are not supported under CMS:

   - The execution of DOS object programs. Although DOS programs can be assembled under CMS (using the VM/370 Assembler), DOS object programs cannot execute under CMS.

   - The writing or updating of OS data sets and DOS files.

6. CMS can read sequential and partitioned OS data sets and sequential DOS files, by simulating certain OS macros.

   The following restrictions apply when CMS reads OS data sets that reside on OS disks:

   - Read-password-protected data sets are not read.

   - VSAM, BDAM, and ISAM data sets are not read.

   - Multi-volume data sets are read as single-volume data sets. End-of-volume is treated as end-of-file and there is no end-of-volume switching.

   - Keys in data sets with keys are ignored and only the data is read.

   - User labels in user-labeled data sets are bypassed.

   The following restrictions apply when CMS reads DOS files that reside on DOS disks:

   - No DOS macros are simulated.

   - Only DOS sequential files can be read. CMS options and operands that do not apply to OS sequential data sets (such as the MEMBER and CONCAT options of FILEDEF and the PDS option of MOVEFILE) also do not apply to DOS sequential files.

   - The following types of DOS files cannot be read:

     --DOS VSAM, DAM, and ISAM files.

     --DOS core image, relocatable, source statement, and procedure libraries.

     --Files with the input security indicator on.

     --DOS files that contain more than 16 user label and/or data extents. (If the file has user labels, they occupy the first extent; therefore the file must contain no more than 15 data extents.)

   - Multi-volume files are read as single-volume files. End-of-volume is treated as end-of-file. There is no end-of-volume switching.

   - User labels in user-labeled files are bypassed.

   - Since DOS files do not contain BLKSIZE, RECFM, or LRECL parameters, these parameters must be specified via FILEDEF or DCB parameters, otherwise, defaults of BLOCKSIZE=32760 and RECFM=U are assigned. LRECL is not used for RECFM=U files.

## Miscellaneous Restrictions

If you intend to run VM/370 Release 1 and Release 2 systems alternately, apply Release 1 PLC 14 or higher (APAR V1179) to your Release 1 system, to provide compatibility and to prevent loss of spool files in case of a warm start.

# Appendix E: Compatibility of VM/370 With CP-67/CMS

VM/370 and its associated component, the Conversational Monitor System, are based on CP-67/CMS and are designed especially for the IBM System/370. Dynamic address translation on the System/370 provides the same facilities as did dynamic address translation (the DAT box) on the System/360 Model 67, but differs in design detail and implementation. Consequently, CP-67/CMS will not run on a System/370 and VM/370 will not run on a System/360. The internal structure and the control blocks of VM/370 differ from the structure and control blocks of CP-67/CMS. User modifications to CP-67/CMS should be re-evaluated considering the new facilities of VM/370 and, if still desirable, redesigned for VM/370.

The Conversational Monitor System of VM/370 functionally extends the Cambridge Monitor System of CP-67/CMS. The following should be fully understood by customers planning conversion to VM/370:

1. The CMS disk file formats are unchanged. CMS provides a utility command to copy files from 2314 or 2319 to 3330, 3333, or 3340, or from the 2311 to the 2314, 2319, 3330, 3333, or 3340. User files on 2311 devices cannot be accessed directly by the Conversational Monitor System and should, therefore, be dumped to tape and then reloaded or copied onto a 2314 or 2319 using the Cambridge Monitor System.

2. The Conversational Monitor System supports ten active disks, with mode letters A-G, S, Y, and Z. The order of search used is alphabetic by mode letter. Existing CMS procedures that use another order of search must be changed.

3. If a CMS command has been issued from a program and that command's format has changed, then the PLIST entries in the program must be modified.

4. The Conversational Monitor System uses VM/370 support for shared segments. For this and other reasons, the CMS nucleus extends beyond location 12000 (hexadecimal). Consequently, all CP-67 MODULE files must be recreated for VM/370.

5. Filename and filetype naming conventions have been changed. Existing file identifications may need to be altered prior to use under the CMS subsystem of VM/370.

6. TXTLIB files must be regenerated because the dictionary format has changed.

For further information, refer to the <u>VM/370</u>: <u>Planning and System Generation Guide</u>.

# Appendix F: VM/370-Related Publications for CMS Users

This appendix provides an index of VM/370-related publications for CMS users. The following VM/370 publications contain general information concerning CMS for new users:

| | |
|---|---|
| VM/370: Quick Guide for Users[1] | GX20-1926 |
| IBM Summary of VM/370 Commands[1] | GX20-1961 |
| VM/370: EDIT Guide | GC20-1805 |
| VM/370: Command Language Guide for General Users | GC20-1804 |

## Corequisite Publications

| | |
|---|---|
| VM/370: Terminal User's Guide | GC20-1810 |
| VM/370: Introduction | GC20-1800 |
| VM/370: EXEC User's Guide | GC20-1812 |
| VM/370: System Messages | GC20-1808 |

## Also Available

| | |
|---|---|
| Virtual Machine Facility/370 Features Supplement | GC20-1757 |
| CMS for Programmers, A Primer | SR20-4438 |

The following sections list, by category of CMS users, the publications that are relevant to the particular CMS user. Since titles change and new publications are constantly being added to the IBM library, this list should serve only as a guide to what is currently available. For a more up-to-date list, see the *IBM System/360 and System/370 Bibliography*, Order No GA22-6822 and the *Virtual Storage Supplement*, Order No. GC20-0001.

*Note*: In some cases, the titles have been abbreviated to save space.

## VS BASIC User

| | |
|---|---|
| VS BASIC CMS Terminal User's Guide | SC28-8306 |
| B is for BASIC. An Introduction to VS BASIC under CMS | SC28-8310 |
| VS BASIC, General Information | GC28-8302 |
| VS BASIC, Program Product Design Objectives | GC28-8301 |
| VS BASIC: Quick Guide for CMS Users | SX28-6386 |
| VS BASIC: Installation Reference Material | SC28-8309 |
| VS BASIC: Language Reference Manual | GC28-8303 |

## BASIC Subroutine User

| | |
|---|---|
| MATH/BASIC, General Information Manual | GH20-1128 |
| MATH/BASIC, Program Reference Manual | SH20-1158 |
| STAT/BASIC, Program Reference Manual | SH20-1069 |
| STAT/BASIC, General Information Manual | GH20-1027 |
| Business Analysis/BASIC, Program Reference Manual | SH20-1264 |
| Business Analysis/BASIC, General Information Manual | GH20-1175 |

## Assembler User

| | |
|---|---|
| OS/VS and VM/370 Assembler Programmer's Guide | GC33-4021 |
| OS/VS, DOS/VS, and VM/370 Assembler Language | GC33-4010 |
| VM/370: System Programmer's Guide | GC20-1807 |

## SCRIPT User

| | |
|---|---|
| SCRIPT/370 Text Processing Facility Under VM/370 - Program Description/Operator's Manual | SH20-1114 |
| SCRIPT/370 IUP: Systems Guide | LY20-0762 |
| SCRIPT/370 Quick Guide for Users Reference Summary | GX20-1969 |

## FORTRAN User

| | |
|---|---|
| VM/370 (CMS) Terminal User's Guide for FORTRAN IV Program Products | SC28-6891 |
| IBM FORTRAN Program Products for OS and the CMS Component of VM/370: General Information | GC28-6884 |
| FORTRAN IV (G1) Code and Go Terminal User's Guide | SC28-6842 |
| IBM OS Code and Go FORTRAN and FORTRAN IV (G1) Programmer's Guide | SC28-6853 |
| FORTRAN IV (G1) Processor and TSO FORTRAN Prompter for OS and VM/370 (CMS): Installation Reference Material | SC28-6856 |

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

[1] These two reference summaries are available separately or can be ordered at the same time by using Order No. GBOF 3576.

| | |
|---|---|
| IBM OS FORTRAN IV Library (Mod 1) for OS and VM/370 (CMS) Installation Reference Manual | SC28-6858 |
| IBM Code and Go FORTRAN Processor for OS and VM/370 (CMS) Installation Reference Material | SC28-6859 |
| IBM OS FORTRAN IV (H Extended) Compiler, Programmer's Guide | SC28-6852 |
| IBM OS FORTRAN IV (H Extended) Compiler and Library (Mod II), Messages | GC28-6865 |
| IBM FORTRAN IV (H Extended) Compiler and FORTRAN Library (Mod II) for OS and VM/370 (CMS) Installation Reference Material | SC28-6861 |
| IBM OS FORTRAN IV Mathematical and Service Subprograms Supplement for Mod I and Mod II Libraries | SC28-6864 |
| IBM FORTRAN IV Library Mathematical and Service Subprograms | GC28-6818 |
| FORTRAN Interactive Debug for OS (TSO) and VM/370-CMS Installation Reference Manual | SC28-6886 |
| FORTRAN Interactive Debug for OS (TSO) and VM/370 (CMS) Terminal User's Guide | SC28-6885 |
| IBM FORTRAN Interactive Debug for OS (TSO) and VM/370 (CMS) Reference Card | SX28-8193 |
| FORTRAN IV Language | GC28-6515 |

COBOL User

Note: "IBM OS Full American National Standard COBOL" has been abbreviated to "OS ANS COBOL" in the list below.

| | |
|---|---|
| OS ANS COBOL Language Manual | GC28-6396 |
| OS ANS COBOL Compiler and Library, Version 4, Programmer's Guide | SC28-6456 |
| OS ANS COBOL Installation Reference Manual | SC28-6458 |
| OS ANS COBOL Messages, Version 4 | SC28-6457 |
| VM/370 OS ANS COBOL Compiler and Library, Version 4 CMS User's Guide | SC28-6469 |
| OS ANS COBOL Version 4 Planning Guide | SC28-6431 |
| OS COBOL Interactive Debug Terminal User's Guide and Reference | SC28-6465 |
| OS COBOL Interactive Debug Installation Reference Material | SC28-6468 |
| OS/VS COBOL Compiler and Library General Information | GC28-6470 |
| OS/VS COBOL Installation Reference Material | SC28-6481 |
| OS/VS COBOL Programmer's Guide | SC28-6483 |

PL/I User

| | |
|---|---|
| OS PL/I Optimizing Compiler, Programmer's Guide | SC33-0006 |
| OS PL/I Optimizing Compiler, Messages | SC33-0027 |
| OS PL/I Optimizing Compiler Execution Logic | SC33-0025 |
| OS PL/I Optimizing Compiler Installation | SC33-0026 |
| OS TSO PL/I Optimizing Compiler | SC33-0029 |
| OS PL/I Transient Library Specifications | GC33-0024 |
| OS PL/I Optimizing Compiler General Information | GC33-0001 |
| OS PL/I Checkout Compiler: General Information | GC33-0003 |
| OS PL/I Checkout Compiler Program Product Specifications | GC33-0030 |
| OS PL/I Checkout Compiler Programmer's Guide | SC33-0007 |
| OS PL/1 Language Reference Manual | GC33-0009 |
| OS PL/I Checkout Compiler Messages | SC33-0034 |
| OS PL/I Checkout Compiler Installation | SC33-0031 |
| OS PL/I Checkout Compiler TSO User's Guide | SC33-0033 |
| OS PL/I Checkout Compiler - Execution Logic | SC33-0032 |
| OS PL/I Checkout Compiler, CMS User's Guide | SC33-0037 |

# Index

IBM Technical Newsletter

| | |
|---|---|
| This Newsletter No. | GN20-2657 |
| Date | March 31, 1975 |
| Base Publication No. | GC20-1800-4 |
| File No. | S370-20 (VM/370 Release 2 PLC 13) |
| Previous Newsletters | None |


# IBM Virtual Machine Facility/370: Introduction



This Technical Newsletter, a part of Release 2 PLC 13 of IBM Virtual Machine Facility/370, provides replacement pages for your publication. These replacement pages remain in effect for subsequent VM/370 releases unless specifically altered. Pages to be removed and/or inserted are listed below.

| | |
|---|---|
| Title page, 2 | 25-26.2 |
| Preface | 31,32 |
| Summary of Amendments | 43-46 |
| Contents | 49-54 (deleted) |
| 9-12 | 57-58 |
| 17-20 | |

Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

SUMMARY OF AMENDMENTS

This Technical Newsletter incorporates changes reflecting support for:

- VM/VS Handshaking
- Remote 3270
- VM/370 Measurement Facility

Note: Please file this cover letter at the back of your publication to provide a record of changes.

IBM Corporation, VM/370 Publications, 24 New England Executive Park, Burlington, Massachusetts 01803

Printed in U.S.A.

Trim Along This Line

## READER'S COMMENTS

**Title:** IBM Virtual Machine Facility/370: Introduction

**Order No.** GC20-1800-4

Please check or fill in the items; adding explanations/comments in the space provided.

Which of the following terms best describes your job?

| | | | |
|---|---|---|---|
| ☐ Customer Engineer | ☐ Manager | ☐ Programmer | ☐ Systems Analyst |
| ☐ Engineer | ☐ Mathematician | ☐ Sales Representative | ☐ Systems Engineer |
| ☐ Instructor | ☐ Operator | ☐ Student/Trainee | ☐ Other (explain below) |

How did you use this publication?

☐ Introductory text ☐ Reference manual ☐ Student/ ☐ Instructor text

☐ Other (explain) ____________________

Did you find the material easy to read and understand? ☐ Yes ☐ No (explain below)

Did you find the material organized for convenient use? ☐ Yes ☐ No (explain below)

Specific criticisms (explain below)

Clarifications on pages ____________________

Additions on pages ____________________

Deletions on pages ____________________

Errors on pages ____________________

Explanations and other comments:

Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

GC20-1800-4

## YOUR COMMENTS PLEASE . . .

This manual is one of a series which serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your comments on the back of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

*Please note:* Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

FOLD FOLD

FIRST CLASS
PERMIT NO. 172
BURLINGTON, MASS.

**BUSINESS REPLY MAIL**
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

**IBM CORPORATION**
**VM/370 PUBLICATIONS**
**24 NEW ENGLAND EXECUTIVE PARK**
**BURLINGTON, MASS. 01803**

FOLD FOLD

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)**

Trim Along This Line