File No. S370-20
Order No. GC20-1800-6

Systems

IBM Virtual Machine Facility/370: Introduction

Release 3 PLC 8

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, a Remote Spooling Communications Subsystem (RSCS), which spools files to and from geographically remote locations, and an Interactive Problem Control System (IPCS), which is a problem reporting process.

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 system requirements, supported language processors and emulators, and VM/370-related publications for CMS users.

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

IBM

Seventh Edition (October 1976)

This is a major revision of GC20-1800-5 and makes obsolete that edition and Technical Newsletters GN20-2677, dated February 16, 1976 and GN20-2687, dated June 18, 1976. This edition corresponds to Release 3 PLC 8 (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/370 Bibliography, 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, Dept. D58, Bldg. 706-2, P.O. Box 390, Poughkeepsie, New York 12602. Comments become the property of IBM.

Preface

This publication introduces and describes the IBM Virtual Machine Facility/370 (VM/370) and its components, the Control Program (CP), the Conversational Monitor System (CMS), and the Remote Spooling Communications Subsystem (RSCS). A fourth VM/370 component, the Interactive Problem Control System (IPCS), is introduced in this publication, but described in detail in the VM/370: Interactive Problem Control System (IPCS) User's Guide, Order No. GC20-1823.

This publication contains four sections and three appendixes:

- "Introduction" describes VM/370, virtual machines, and their applications.
- "Control Program" 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.
- "Appendix A: System Requirements"
- "Appendix B: Language Processors and Emulators"
- "Appendix C: VM/370-Related Publications for CMS Users"

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

The term 2305 refers to the IBM 2305 Fixed Head Storage, Models 1 and 2.

When the term 3330 is used in this publication, it refers to the IBM 3330 Disk Storage Models 1, 2, and 11; the IBM 3333 Disk Storage and Control, Models 1 and 11; or the IBM 3350 Direct Access Storage operating in 3330/3333 Model 1 or 3330/3333 Model 11 compatibility mode.

The term 3340 refers to the IBM 3340 Disk Storage, Models A2, B1, and B2 and the IBM 3344 Direct Access Storage Model B2.

The term 3350 refers to the IBM 3350 Direct Access Storage, Models A2 and B2.

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

Information on the IBM System/370 Models 135-3, 138, 145-3, and 148 is for planning purposes only until the availability of the product. Unless otherwise stated, references to the System/370 Models 138 and 148 also apply to Models 135-3 and 145-3 respectively.

RELATED PUBLICATIONS

IBM Virtual Machine Facility/370:

CMS Command and Macro Reference, Order No. GC20-1818

CMS User's Guide, Order No. GC20-1819

Commands (General User) Order No. GX20-1961

Commands (Other than General User), Order No. GX20-1995

CP Command Reference for General Users, Order No. GC20-1820

Environmental Recording, Editing, and Printing (EREP) Program, Order No. GC29-8300

Glossary and Master Index, Order No. GC20-1813

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

Operating Systems in a Virtual Machine, Order No. GC20-1821

Operator's Guide, Order No. GC20-1806

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

Quick Guide for Users, Order No. GX20-1926

Release 3 Guide, Order No. GC20-1822

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

System Messages, Order No. GC20-1808

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

Features Supplement, Order No. GC20-1757

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

Data Areas and Control Block Logic, Order No. SY20-0884

Environmental Recording, Editing, and Printing (EREP) Program Logic, Order No. SY25-7701

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

System Logic and Problem Determination Guide, Order No. SY20-0885

Note: The VM/370: Quick Quide for Users, VM/370: Commands (General User) and VM/370 Commands (Other than General User) 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 are given in abbreviated form.

SUPPLEMENTARY PUBLICATIONS

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

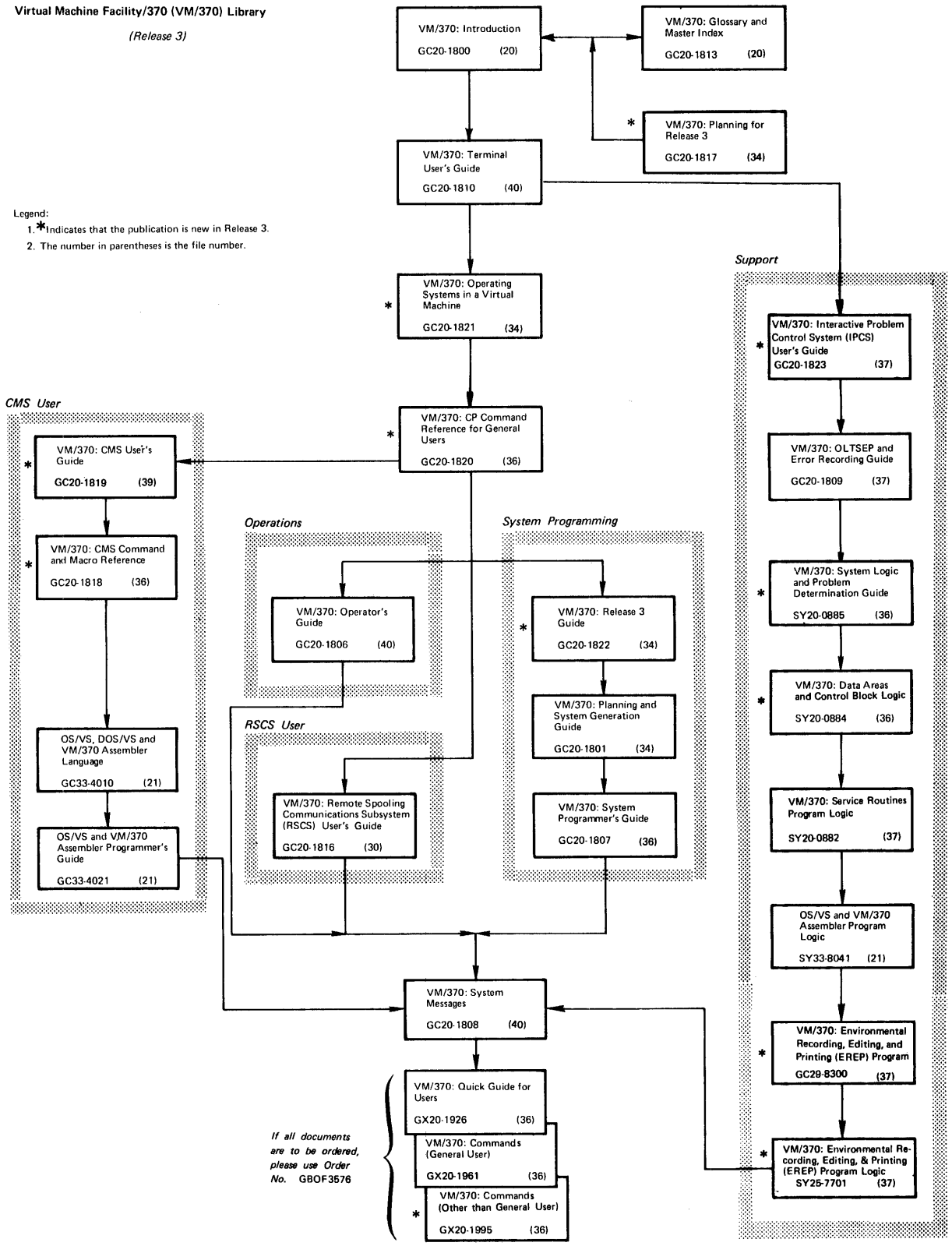

Figure 1. Virtual Machine Facility/370 Library

Contents

Figures

VM/370 SUPPORTS THE VIRTUAL MACHINE COMMUNICATION FACILITY (VMCF)

New: Program Feature

The Virtual Machine Communication Facility (VMCF) provides a method of communication and data transfer between virtual machines operating under the same VM/370 system. Data is transferred in up to 2048-byte blocks from virtual storage to virtual storage; data length is limited only by the sending and receiving machine storage sizes.

VMCF is described in "Virtual Machine Communication Facility."

VM/370 SUPPORTS THE NEW SYSTEM/370 MODELS 135-3, 138, 145-3, AND 148

New: Hardware Feature

VM/370 supports the System/370 Models 135-3, 138, 145, and 145-3. This support includes the following:

- The System Display Console for the above CPUs.

  In display mode the system console is supported on a 3277 display terminal. In printer-keyboard mode, the 3286 console printer is required.

- The 3203 Printer Model 2 (for the System/370 Models 138 and 148 only).

- The virtual machine assist function

- VM/370 Extended Control-Program Support

VM/370 Extended Control-Program Support is a combination of a CPU hardware assist and VM/370 programming. This support is described in "VM/370 Extended Control-Program Support" of the "Control Program" section.

VM/370 SUPPORTS THE IBM SYSTEM/32 AND SYSTEM/3 FOR RSCS

New: Device Support

The System/32 and the System/3 Models 6, 8, 10, 12, and 15, with the MULTI-LEAVING[1] Job Entry Work Station (MRJE/WS) System Utility Program are now supported as RSCS remote work stations. "Appendix A: System Requirements" reflects this support.

VM/370 NOW SUPPORTS THE IBM SYSTEM/370 MODEL 168 ATTACHED PROCESSOR SYSTEM

New: Specification Change

The IBM System/370 Model 168 Attached Processor System is now supported. "Appendix A: System Requirements" reflects this support.

---

[1]Trademark of IBM

INTERACTIVE PROBLEM CONTROL SYSTEM (IPCS) SUPPORTED AS A VM/370 COMPONENT

New: Program Feature

The Interactive Problem Control System (IPCS) provides online problem tracking; it uses facilities using an extended form of the CMS VMFDUMP command and five new CMS commands. IPCS runs as a service program in a CMS virtual machine. For detailed information describing IPCS, refer to the publication VM/370: Interactive Problem Control System (IPCS) User's Guide, Order No. GC20-1823.

VS/APL SUPPORTED BY CMS

New: Program Feature

CMS supports the VS APL interpreter. This support is reflected by the following changes to this publication:

- "Language Processors" in the "Conversational Monitor System" section is updated.
- "Appendix B: Language Processors and Emulators" is updated.
- "Appendix C: VM/370-Related Publications for CMS Users" is updated.

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 the IBM System/370 Models 135, 135-3, 138, 145, 145-3, 148, 155 II, 158, 165 II, and 168. The real System/370 must have the Dynamic Address Translation (DAT) feature, a hardware feature 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:

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

A major difference between CP-67/CMS and VM/370 is that VM/370 has 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, the IBM 3350 Direct Access Storage and the IBM 2305 Fixed Head Storage, and offers several options to enhance performance in the virtual machine environment.

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

VM/370 has four components:

- The Control Program (CP): CP controls the resources of the real computer to provide multiple virtual machines.
- The Conversational Monitor System (CMS): CMS users have a wide range of conversational, time-sharing functions. CMS users can create and manage files, and compile, test, and execute problem programs.
- The Remote Spooling Communications Subsystem (RSCS): RSCS users can transmit files to and receive files from remote stations in the RSCS teleprocessing network.
- The Interactive Problem Control System (IPCS): IPCS provides VM/370 problem analysis and management facilities, including problem report creation, problem tracking, and CP abend dump analysis.

Virtual Machine Operating Systems

While the control program of VM/370 manages the concurrent execution of the virtual machines, an operating system must manage 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.

Figure 2 lists the operating systems that can execute in virtual machines. CP provides each of these with virtual device support and virtual storage. The operating systems themselves execute as though they are controlling real devices and real storage, but they must not violate any of the VM/370 restrictions listed in the VM/370: Planning and System Generation Guide. For more information about virtual machines, see VM/370: Operating Systems in a Virtual Machine.

Batch or Single-User Interactive	Multiple-Access
DOS	VM/370
DOS/VS	Time-Sharing
OS/PCP	Option of OS
OS/MFT	
OS/MVT	
OS/VS1	
OS/VS2	Conversational
OS-ASP	CMS
PS44	
RSCS	

Figure 2. Virtual Machine Operating Systems

Figure 3 shows six virtual machines executing concurrently under the control of CP on an IBM System/370 Model 138, with 1024K of real storage. One machine is doing batch production work under the current release of DOS; a second is executing programs that require a previous release of DOS; and a third is controlling the RSCS network. The other three virtual machines are executing CMS: there is one virtual CMS machine for each of three separate conversational users.

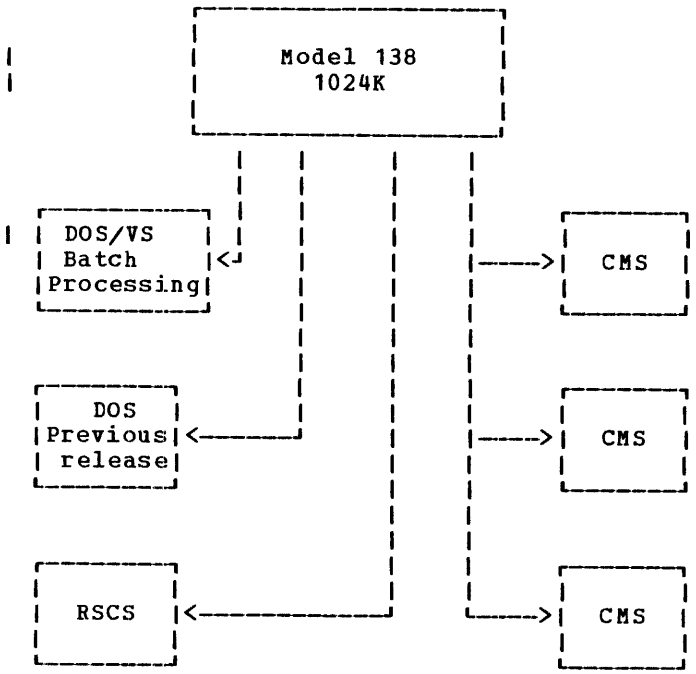

Figure 3. Multiple Virtual Machines

The VM/370 Directory

The VM/370 directory is a disk-resident file that contains an entry for each of 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 command 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 use his virtual machine.

Figure 4 shows a logon procedure of a user whose userid is "smith." When he enters his password, the printing or displaying of the password may be masked for security purposes. 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 THURSDAY 01/15/76

ipl cms
```

Figure 4. Logging on VM/370 and Loading 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 virtual machine components follow.

VIRTUAL SYSTEM CONSOLE

The user's terminal, for example an IBM 2741 Communication Terminal or an IBM 3277 Display Station, serves as the virtual system console. By entering commands at his terminal, a user can perform almost all the functions an operator can perform on a real machine system console. He can load an operating system, stop and start virtual machine execution, and display and change the contents of registers and storage.

VIRTUAL STORAGE

Each virtual machine has its own virtual storage space. It 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. The virtual machine uses this storage space exactly as though it is real storage, but it is not limited by the storage size of the real machine. For example, three virtual machines of 256K bytes each can execute 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, but does not necessarily keep in storage those parts that are not needed immediately. Instead, they may be sent to a direct access device and stored until they are needed again.

The virtual storage size defined in the VM/370 directory may vary for virtual machines.

Each virtual machine can refer only to its own virtual storage. Thus each virtual machine's storage is protected 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, each virtual CPU has available the facilities described in IBM System/370 Principles of Operation, Order No. GA22-7000. Some restrictions exist; they are discussed in the VM/370: Planning and System Generation Guide.

Single task or multitask operating systems can execute in a virtual machine. For example, both CMS (a single task operating system) and OS/VS1 (a multitask operating system) can execute in VM/370 virtual machines.

The virtual CPU can execute in either basic or extended control mode (extended control mode includes all the facilities necessary to execute VM/370 as the virtual machine's operating system). For example, OS/MFT and OS/VS1, as well as CMS and VM/370, can execute 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 supports the same devices as a real machine. Virtual devices are logically controlled by the virtual machine and not by VM/370. In most cases, input/output (I/O) operations, and any error recovery processing, are 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 any data translations that are necessary.

All virtual devices must have real counterparts. For example, a virtual disk must have a real disk counterpart, or a virtual tape must have a real tape counterpart. Some virtual devices, such as tapes, must have a one-to-one relationship with a real device. Others may be assigned a portion of a real device (for example, a virtual disk may occupy all or part of a real disk). Several virtual disks may be assigned to one real disk.

Figure 5 shows three virtual disks that are assigned to one real 2319 disk volume. These virtual disks may belong to three different virtual machines or to one virtual machine; they can be used by CMS, DOS, or OS. Virtual disks are also called minidisks because more than one can be assigned to a full-sized real disk. VM/370 distributes service programs which create and change minidisks.

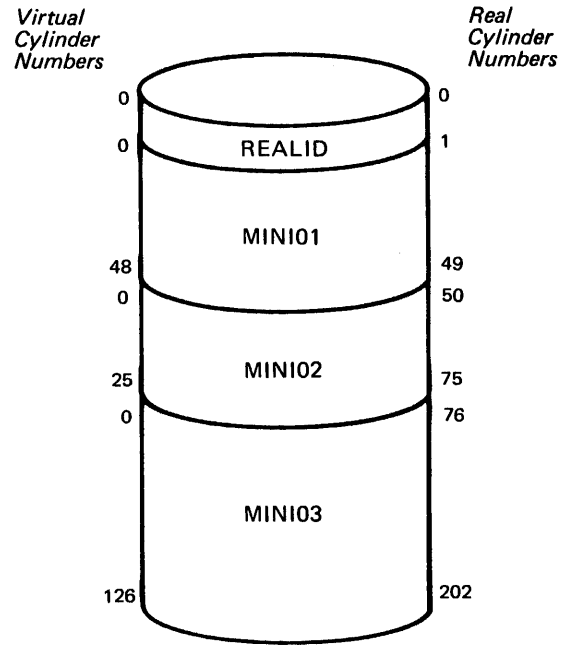

Figure 5. Real Disk Containing Minidisks

In Figure 5, the volume serial number of the real disk is REALID. MINI01, MINI02, and MINI03 are the labels of minidisks on the real 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. Usually a real unit record device is not kept busy constantly, thus the input and output of several virtual unit record devices can 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 storage space that has been allocated for this purpose. This function is called spooling and is explained under "Spooling Unit Record I/O" in the "Control Program" section.

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. To dedicate a virtual device, either designate it as dedicated in the VM/370 directory or have it attached to a specific virtual machine.

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

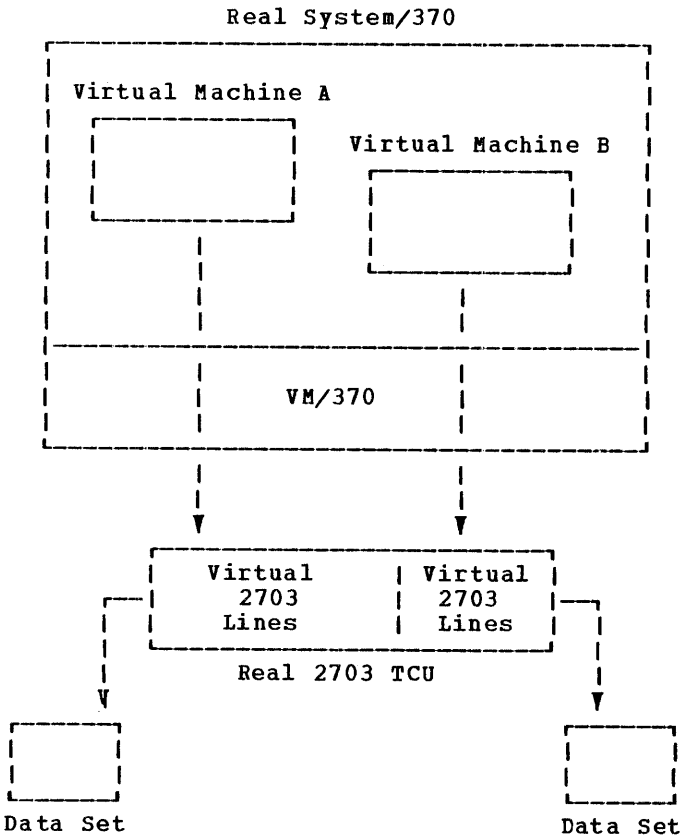

Figure 6. Virtual Devices: Transmission Control Units

A virtual channel-to-channel adapter (CTCA) can be defined either with or without a real equivalent. If a real channel-to-channel adapter exists, a virtual machine can communicate with a real computing system other than its own; if it does not exist, a virtual machine can only communicate with virtual machines in the same computing system.

An alternative to a virtual channel-to-channel adapter is the Virtual Machine Communication Facility (VMCF), described in the "Control Program" section of this publication.

Restrictions that apply to virtual machines are discussed 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 the performance, are described under "Performance Options For Virtual Machine Time Management," "Performance Options For Virtual Machine Storage Management," and "Performance" in the "Control Program" section.

VM/370 Applications

Using VM/370, an installation can perform its work more efficiently and easily. Virtual machine applications aid in programming, operations, and interactive use.

PROGRAMMING

The virtual machine environment lends itself to program development because:

- Programs developed in a virtual machine can exceed the real storage size of the real computer.
- 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.
- JCL (Job Control Language) is not needed when compiling, assembling, and/or testing under CMS.
- Users can test privileged programming code in their own virtual machines.
- Programmers can use debugging aids at their terminal that parallel those of an

operator at a system console: they can (1) display and store into the general or floating-point registers or into virtual storage, (2) stop execution at an instruction address, and (3) alter the normal flow of execution. The 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.")

- 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.

- RSCS users can transmit files to and receive files from users at other geographic locations.

- 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.

- 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. (For a more complete discussion of program execution under CMS refer to "Program Development and Execution" in the "Conversational Monitor System" section.) DOS assembler language programs can be compiled under CMS if the installation adds the appropriate DOS macros to the CMS system.

OPERATIONS

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

- System generation, support, and testing of operating systems, as well as conversion, can be done without a dedicated real machine, concurrently with normal production work. Thorough testing in a virtual machine reduces errors and the possibility of abnormal terminations of the system. For example, a user can apply a program temporary fix (PTF) to an IBM operating system and test that system in one virtual machine while he does regular production work in another virtual machine using the same IBM operating system without the PTF applied. This concurrent operation can be done provided sufficient direct access storage resources are available. The virtual machine test is analogous to one made on a real machine, provided that:

  - There are no timing dependencies.

  - The test is not measuring time.

  - Dynamically modified channel programs are not used except as noted in VM/370: Planning and System Generation Guide.

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

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

- Many types of batch applications can be executed, either in an individual virtual machine or in a virtual machine dedicated to executing programs in batch mode, with no change to the program.

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

- 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 used on another System/370 if the device addresses on both machines are the same. This is not unique to VM/370; the same procedure is used to back up OS or DOS systems.

  - This method is unique to VM/370. The volumes containing only user minidisks may be carried to another computing system that is using VM/370. When this method is used, the VM/370 directory on the backup system must be updated to include the virtual machines that it must now support.

    The backup system must include, but is not limited to, the same type and number of real devices as these virtual machines require. Also,

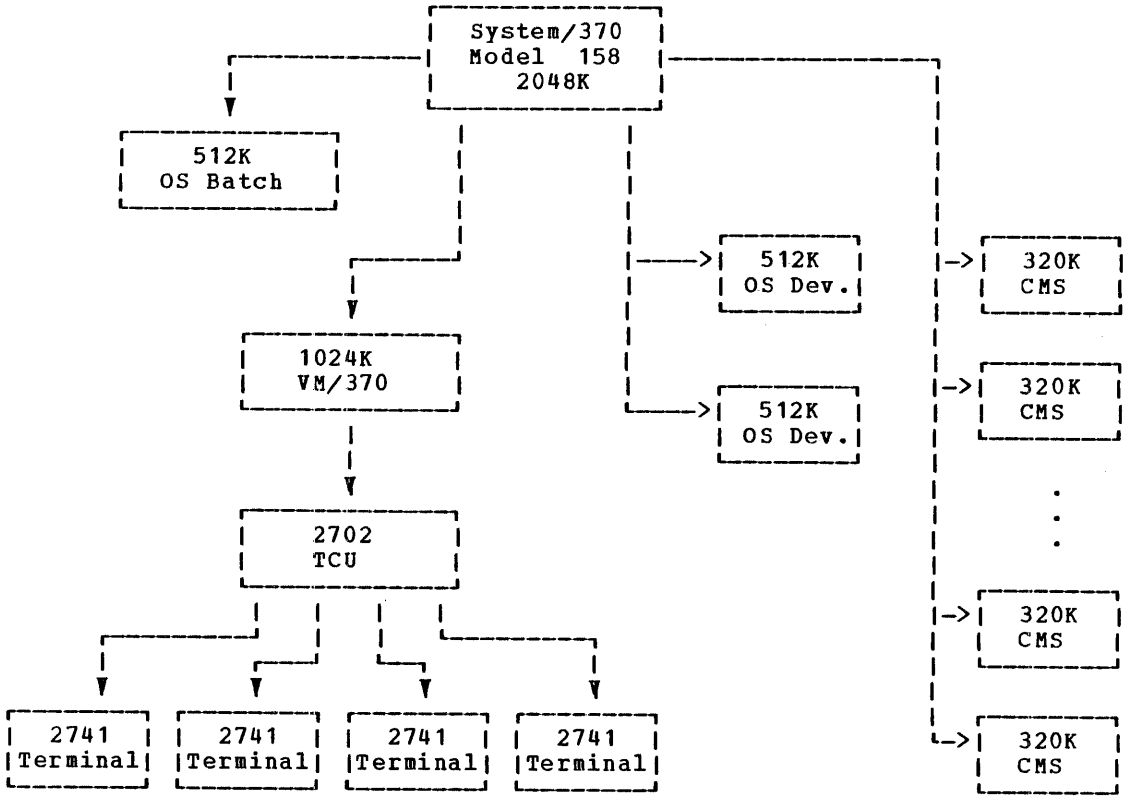

Figure 7. Virtual Machines for Concurrent Production, Development, and Testing

the backup system must have a sufficient number of direct access storage drives so that the user volumes can be mounted.

Because the virtual devices defined for the virtual machines 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 assigns real devices just as it does for its own virtual machines. Thus, the production work of the system being backed up can be executed in virtual machines concurrently with the execution of the virtual machines of the backup system.

INTERACTIVE USE

Two kinds of interactive systems execute under VM/370: multiple-access and single user.

- Multiple-access systems like VM/370 execute 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.

  Once his terminal is connected, the user issues only the commands associated with the multiple-access system.

  For example, the DIAL command connects the user's terminal with a VM/370 system executing in a virtual machine under VM/370. Once his terminal is connected, the user communicates only with that particular version of VM/370.

  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.

- Systems that can be executed interactively by a single user include the Conversational Monitor System and any operating system that can execute in a virtual machine. A time-sharing environment is created when VM/370 creates multiple virtual machines, each controlled by the same operating system. These systems operate concurrently with each other as well as with other conversational or batch systems.

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

When a user logs on VM/370, CP creates a virtual machine for him. Based on information stored in the VM/370 directory, CP creates a virtual machine with a specific amount of virtual storage and specific virtual devices. The command privilege classes allowed for the virtual machine and optional support (such as, extended control mode) are also determined by each virtual machine's entry in the VM/370 directory.

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

VM/370 uses a technique called time slicing so that one real CPU appears to be multiple virtual CPUs. Each virtual machine periodically gains access to the real CPU for a small amount of time, called a time slice. CP determines how frequently and for how much time a virtual machine gains access to the real CPU by examining the number of console requests, or terminal interrupts, the virtual machine has issued during its past time slices. If the number is large, CP defines the virtual machine as a conversational user and assigns it the smaller of two possible time slices. If the number is small, the virtual machine is a nonconversational user and is assigned the larger time slice. CP gives conversational users more frequent access to the real CPU for short time slices, while it gives nonconversational users larger time slices at less frequent intervals.

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 loaded from auxiliary storage into real storage
- An input/output operation to be translated, begun, or completed
- A CP command to finish executing

PERFORMANCE OPTIONS FOR VIRTUAL MACHINE TIME MANAGEMENT

VM/370 has options that can be used to improve the performance of a virtual machine. A virtual machine with a performance option is termed preferred. Although each option can be applied to a different virtual machine, if optimum performance is required, usually performance options are assigned for only one or two specific virtual machines.

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

A priority value assigned to a virtual machine is used, in combination with other factors, to influence the dispatching algorithm. A low priority value gives that virtual machine a larger slice of CPU time, provided that virtual machine can fully utilize the time. The priority option 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 specified in 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.

For more detailed information on these and other options that improve virtual machine performance, see 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). When a user logs on, his virtual machine storage size is the amount of storage defined as the 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 virtual machine's maximum size.

Storage in the virtual machine is logically divided into 64K (65,536) byte areas called segments. These are further divided into 4K byte areas called pages. For each virtual machine, CP creates and updates 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, to optimize real storage usage only virtual storage pages that are referred to frequently are kept in real storage. 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 stored on a direct access storage device. If an inactive page is changed at some time during virtual machine execution, CP assigns it to a paging device, selecting the fastest paging device with available space. If the page is 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 one or more specific pages of virtual storage is no longer needed; CP then releases the areas of the paging device that were assigned to hold the specified pages.

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 an operating system can create and control its own virtual storage, 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 creates and updates a set of page and segment tables that relate this virtual storage to the virtual storage of the virtual machine. In VM/370, "first level storage" refers to real storage, "second level storage" refers to virtual machine storage, and "third level storage" refers to the virtual storage created and controlled by the virtual machine. When

OS/VS1 is executing, instructions and data from third level storage must be available to the CPU. Thus 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 updates such tables, called shadow tables. CP has 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 reenterable 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 CP-owned volume.

If a user that is executing a shared system alters one of the shared segments, his system is put into nonshared mode. Thus, whenever a user issues an ADSTOP, STORE, TRACE INSTRUCT, or TRACE BRANCH command that alters a shared segment, the shared system containing that segment is placed in nonshared mode.

Noncontiguous segments can be attached to and detached from virtual machines. They can be shared or nonshared. Noncontiguous segments may be within the virtual machine's defined storage, appended to the end of its virtual storage, or loaded at addresses beyond its virtual storage. VM/370 supports noncontiguous segments for CMS; in this case, the addresses of the noncontiguous segments must be greater than the highest address in the virtual machine that is attaching them. For a description of shared segments, noncontiguous segments, and named systems see the VM/370: System Programmer's Guide.

PERFORMANCE OPTIONS FOR VIRTUAL MACHINE STORAGE MANAGEMENT

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 a certain number of page frames to a specified virtual machine. 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 assign the virtual=real option to one or more virtual machines. However, only one virtual machine can use the virtual=real area at any one time. With this option, a virtual=real area is allocated directly from real storage when VM/370 is initially loaded, and that area remains allocated unless it is released by the system operator. All pages, except the virtual machine's page zero, are allocated to the corresponding real storage locations. (To control the real computing system, CP must control real page zero.) Consequently, the real storage size must be large enough to accommodate the CP nucleus, the entire virtual=real area, 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. Also, noncontiguous segments are not allowed for virtual=real systems. The VM/370: Planning and System Generation Guide discusses some situations in which the virtual=real option is necessary, and contains the formulas needed to determine the amount of virtual=real storage that is available for varying real storage sizes.

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

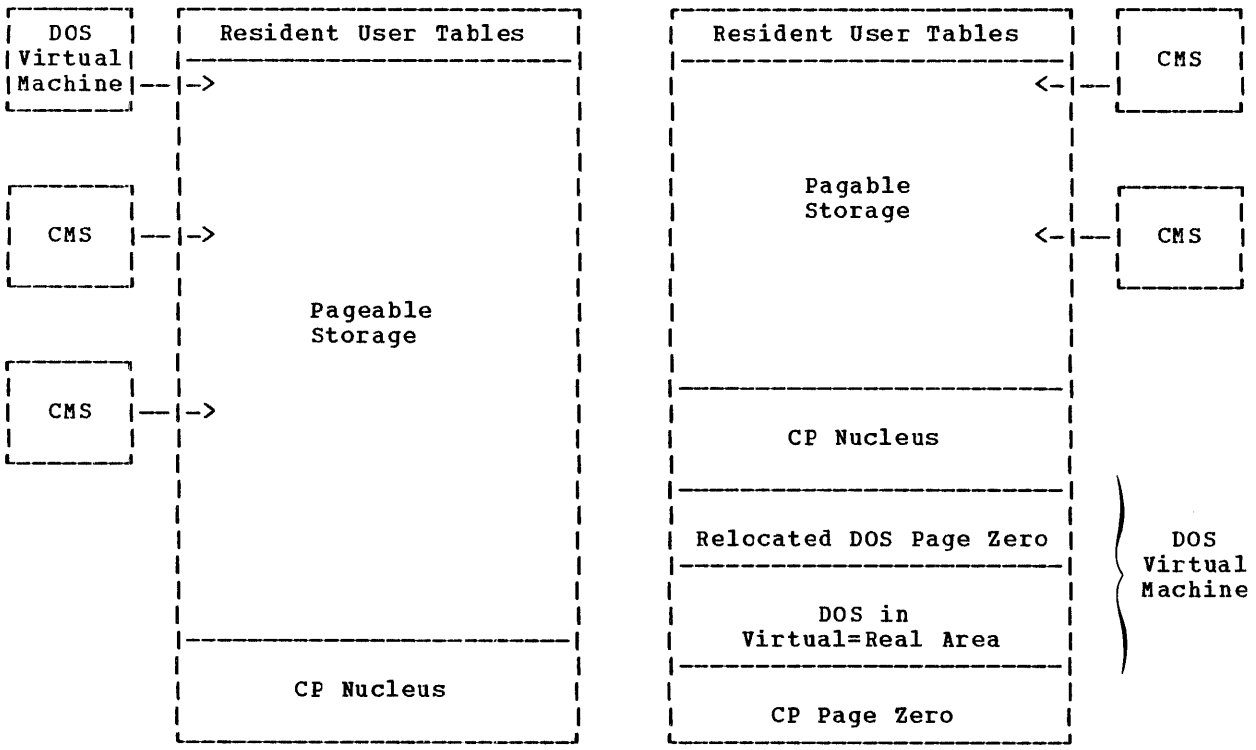

Figure 8. Comparison of Storage Usage for Virtual Machines With and Without Virtual=Real Option

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 VM/370 directory entry of the virtual machine, or they may be attached to (or detached from) the virtual machine while it remains logged on. Virtual devices may be dedicated, if they are assigned to a fully equivalent real device; shared, if a minidisk is linked by more than one virtual machine; or spooled by CP to intermediate direct access storage.

When OS executes in a real machine, input/output operations are 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 referred to are virtual. CP translates the virtual addresses to real addresses.

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

During this processing, CP designates the virtual machine as not executable. When the virtual machine gains control, CP gives it a suitable condition code (as on a real machine) to indicate the status of the Start I/O operation. In addition, CP reflects the interrupts caused by the input/output operation 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 VM/370: OLTSEP and Error Recording Guide.

The programs to be executed 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.

Virtual disks can be shared by several virtual machines. Virtual disk sharing is specified in the VM/370 directory entry or by a user command. If the user issues the CP LINK command to share a virtual disk, he must supply the appropriate password before he can gain access to the virtual device.

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

Virtual disks 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 disk.

A virtual machine may be assigned a dedicated channel, via the ATTACH CHANNEL command. If a virtual machine is assigned a dedicated channel, it has that channel and all of its devices for its exclusive 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 by any other virtual machine. Virtual machines may have a mixture of dedicated and nondedicated channels.

A virtual input/output operation by CP can be simplified if the virtual machine uses the Diagnose interface. The Conversational Monitor System, which was 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 the Diagnose interface 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.

For a description of how virtual machines running under the same VM/370 system can communicate and exchange data, see "Virtual Machine Communication Facility" in this section.

Spooling Unit Record I/O

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

CP, not the virtual machine, controls the unit record devices that are 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 from real card decks. The real machine operator places the card deck in the input hopper of the real card reader. The real card deck must be preceded by a USERID card that names the virtual machines to receive the card deck.

Input spool files can also be created by using RSCS remote work stations. The operator at the remote work station must place a card deck in the hopper of the remote card reader. This card deck must be preceded by a USERID card that names the RSCS virtual machine and the virtual machine that is to receive the card file. 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 CLOSE command.

If the direct access storage space assigned to spooling becomes full, spooling stops and the virtual unit record devices appear to be not ready. The spooling operator must make additional spooling space available. He can purge existing spool files or assign additional direct access storage space for spool files.

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-character records.) Files transferred between virtual unit record devices by the spooling routines are not physically punched or printed. The CP spooling support can make files available to multiple virtual machines, or to different operating systems executing at different times in the same virtual machine.

CP can print multiple copies of a single spool file, backspace any number of printer pages, and define spooling classes for real output files.

Spooling Virtual Console I/O

CP allows the user to spool his virtual machine's console input/output 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. Console spooling is invoked by the SPOOL CONSOLE command. It is particularly useful when the virtual machine is executing with the terminal disconnected, because the virtual console output, which would otherwise be lost, is saved on disk. The saved data is later printed on the real printer. When a console spool file is closed, it becomes a printer spool file.

Remote Spooling

CP, in conjunction with RSCS, supports remote spooling, that is, RSCS transmits files across a teleprocessing network. The "Remote Spooling Communications Subsystem" section describes RSCS and how it is used.

CP Commands

CP commands are used interactively by operators and systems personnel to control the real computing system and VM/370, and by users to control virtual machines and their operating systems.

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 signaling an attention interrupt to VM/370's control program; a virtual machine attention interrupt is equivalent to 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 signaling 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. The privilege classes define the subset of CP commands that each user can execute.

Figure 9 defines each privilege class, listing the functions and the users associated with each privilege class. Figure 9 also indicates which publications describe the commands for each class.

GENERAL USERS

To activate his virtual machine, the user must establish a connection with the real computing system that is executing VM/370. At that point, the user issues the LOGON command to identify himself to VM/370. VM/370 then creates the control blocks necessary to simulate the virtual machine configured in the user's VM/370 directory entry.

The user's terminal can also be a remote terminal for a multiple-access virtual machine operating system (such as VM/370). To identify himself to a multiple-access system, the user must issue the DIAL command. Thereafter, his terminal is controlled by the multiple-access system directly. The user's terminal must be of a type supported by the multiple-access system.

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 during IPL. 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. 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]	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[3]	General User: The class G user controls functions associated with the execution of his virtual machine.
Any [3]	The Any classification is given to certain CP commands that are available to any user. These are primarily for the purpose of gaining and relinquishing 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 Logic and Problem Determination Guide.
[3]Described in the VM/370: CP Command Reference for General Users.

Figure 9. CP Privilege Class Descriptions

After the user has his terminal connected, he loads an operating system into his virtual machine, using IPL command. The user can stop execution in the virtual machine to invoke commands that simulate the functions of the operator's console or control his virtual machine. Some of these commands are:

Command	Meaning
ADSTOP	Defines an instruction address stop location in virtual storage.
BEGIN	Resumes execution in the virtual machine (the functional equivalent of pressing the Start key on a real computing system).
DETACH	Removes a specified device from the virtual machine configuration.

Command	Meaning
DISPLAY	Displays specified virtual machine registers or virtual storage contents in hexadecimal or EBCDIC.
EXTERNAL	Causes an external interrupt.
LINK	Makes a specified virtual direct access storage device a part of the virtual machine configuration if the device is defined as shared and the user can supply the appropriate password.
QUERY	Displays certain system information such as the log message, the number of spool files, or the virtual machine configuration.
READY	Simulates a device end interrupt from a virtual device.
SET	Establishes certain system values such as the level of error message to be printed or the amount of VM/370 line editing for terminal input lines.
SPOOL	Alters the spooling control options (such as number of copies) for one or more virtual unit record devices that are used for spooling. 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.
TERMINAL	Allows the user to define the VM/370 logical editing symbols and the logical line size of I/O to and from his terminal.

OTHER USERS

Users, other than the general user, can perform additional functions. 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 during VM/370 system generation and specified in the VM/370 directory for selected virtual machines.) System operators control the orderly activity of the real computing system with the FORCE command to terminate a particular virtual machine, SHUTDOWN to terminate all VM/370 functions so that a VM/370 restart can be performed, and ACNT to create accounting records for all active users.

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. The SAVESYS command saves an image of a virtual machine's storage space, registers, and program status word on a CP-owned volume.

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 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 or the VM/370 Extended Control-Program Support.

In general, the best performance is seen in a VM/370 system that has a large amount of real storage, VM/370 hardware assist,

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 virtual machines using CMS than, for example, virtual machines using OS.

It may be possible to improve the performance of an operating system such as OS by substituting CP paging for virtual machine I/O operations. To do this, the user must specify frequently-used OS functions (such as, transient subroutines and ISAM indexes) as resident. 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 (Start I/O) operations are simulated by CP in three channel modes: byte multiplexer, selector, and block multiplexer.

Virtual byte multiplexer channel mode is reserved for I/O operations for 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. Because the CTCA is treated as a shared control unit, it must be connected to a selector channel.

Block multiplexer channel mode allows the virtual machine's operating system to overlap SIO requests to multiple devices connected to the same channel. For a virtual machine in block multiplexer mode, CP simulates a real block multiplexer operation.

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 in the VM/370 directory entry for a virtual machine.

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 execute 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. The virtual machine assist feature is standard on the System/370 Models 135-3, 138, 145-3, and 148.

Whenever VM/370 is loaded 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 SASSIST 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.

Under some conditions the virtual machine assist feature cannot be used. CP automatically turns the feature off if the user invokes certain TRACE functions. CP automatically turns the feature on again when the user ends the TRACE FUNCTION.

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.

VM/370 EXTENDED CONTROL-PROGRAM SUPPORT

VM/370 Extended Control-Program Support is a hardware assist function that is available only on the System/370 Models 135-3, 138, 145-3, and 148. This hardware assist function, when used with the standard virtual machine assist feature

described previously, further reduces VM/370's real supervisor state time needed to support virtual machines. VM/370 Extended Control-Program Support provides the following functions:

- Expanded Virtual Machine Assist
- CP Assist
- Virtual Interval Timer Assist

Whenever VM/370 is loaded on one of the above CPUs, all three hardware assist functions plus virtual machine assist are activated unless turned off by the system operator, or unless the hardware level and software level do not match.

Expanded virtual machine assist includes a more comprehensive emulation of the SSM, LPSW, STNSM, and STOSM privileged instructions. Additional privileged instructions are also emulated. The virtual machine assist function is not part of expanded virtual machine assist.

CP assist provides a hardware assist for the high-use portions of the following CP functions:

- Virtual Machine I/O
- Storage Management
- Page Management
- Privileged Instruction Handler
- Dispatcher

The appropriate CP software routine is used if (1) CP assist is turned off, (2) assist feature does not support the specific service required, or (3) an error condition occurs.

Virtual interval timer assist provides for hardware updating of the location 80 interval timer for each virtual machine that has the virtual timer assist function turned on. This timer assist provides a more accurate and repeatable interval timer value for virtual machines than was previously possible through CP software.

Both virtual machine assist and expanded virtual machine assist are automatically turned off if the user invokes certain TRACE functions. In addition, virtual interval timer assist is turned off if external interrupts are traced. When the tracing function is terminated, CP automatically reactivates these VM/370 hardware assist functions.

For more details on VM/370 Extended Control-Programm Support, 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: CP Command Reference for General Users for details about the MONITOR and INDICATE commands.

VIRTUAL MACHINE COMMUNICATION FACILITY

The Virtual Machine Communication Facility (VMCF) provides the capability for one virtual machine to communicate with and exchange data with any other virtual machine operating under the same VM/370 system.

Messages and data are directed to other virtual machines via the userid. Data is transferred in up to 2048 byte blocks from the sending virtual machine's storage to the receiving virtual machine's storage. The amount of data that can be moved in a single transfer is limited only by the sizes of virtual machine storage of the respective virtual machines.

Use of real storage is minimal. Only one real storage page need be locked during the data transfer. A special external interrupt is used to notify one virtual machine of a pending transfer of data; this interrupt is also used to synchronize the sending and receiving cf data.

VMCF is implemented by means of functions invoked using the DIAGNOSE instruction and a special parameter list.

For example, the SEND function directs a message or block of data from virtual machine storage in the source (or sending) virtual machine to virtual storage in the sink (or receiving) virtual machine; the RECEIVE function allows a virtual machine to selectively accept messages or data.

A more detailed description of VMCF functions and how they can be invoked in a virtual machine is contained in the VM/370: System Programmer's Guide. A description of VMCF logic is contained in the VM/370: System Logic and Problem Determination Guide.

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:

- Closes CP spool files
- Processes VS1 pseudo page faults
- Provides an optional nonpaging mode for VS1 when it executes with VM/370
- Provides miscellaneous enhancements for VS1 when it executes 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 interrupt 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 interrupt 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 executes 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 executes in nonpaging mode than when it does not. Nonpaging mode is available only when the VM/VS Handshaking feature is active.

MISCELLANEOUS ENHANCEMENTS

When OS/VS1 executes in a VM/370 environment, some 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
- 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 be generated with the VM/370 option and execute with a version of VM/370 that supports VM/VS Handshaking. In addition, VS1 must execute in Extended

Control (EC) mode and its virtual machine size must not exceed 4M.

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: CP Command Reference 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 through:

- Multiple and separate virtual machines (see "RAS Features of VM/370 Design").
- The use of the RAS features of the System/370 hardware and VM/370 system control programming. These are discussed in detail in the VM/370: OLTSEP and Error Recording Guide.

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.

- 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.
- Passwords must be used to gain access to the system and to shared disk files.
- The data on a shared or critical disk file can be protected by designating it as a read-only disk.
- If a virtual machine abnormally terminates, only that machine and its user (or users, if it is a multiple-access system) are affected.
- Users can run concurrently as many versions, levels, types, and copies of operating systems as they require. Systems can be generated and tested in one virtual machine while production work is done in another. A new system can be fully tested before conversion with no impact on the production work schedule and no need for a real machine dedicated to testing.
- VM/370 has commands to trace, examine, and alter the operation of a virtual machine and to examine and alter the contents of its virtual storage. These commands are useful for program debugging.
- 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 in other virtual machines.

The Conversational Monitor System (CMS) is a component 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 has 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 multitasking (multiprogramming) capabilities, as it is designed to execute in a VM/370 virtual machine. CP provides the time-sharing environment; CMS provides the conversational user interface. Using CMS, the user can write programs to be executed under CMS or another virtual machine operating system.

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 may not be in real storage at any given time. However, the most active pages tend to remain in real storage.

Additional portions of CMS can be shared in discontiguous segments. Discontiguous segments can be attached to and detached from the CMS virtual machine as needed. A name is associated with one or more discontiguous segments. Some, all, or none of the discontiguous segments can be shared. Discontiguous segments must be loaded at addresses beyond the highest address in the virtual machine.

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 because CMS has no unit record error recovery procedures.

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

Generally, each CMS user is assigned at least two disks: a read-only system disk and a read/write user disk.

The read-only system disk contains the CMS nucleus, disk-resident CMS commands, and the system library. The CMS system disk can be shared among CMS users.

The read/write disk contains the user's permanent and temporary files. The size of a CMS user disk is limited to one volume or the maximum number of CMS records that can be contained on one volume. CMS disks must be assigned in units of full cylinders. The maximum size of CMS disks, by device types, is:

Device	Maximum Size CMS Disk, in Cylinders
2314/2319	203
3330	246
3340-35	349
3340-70	682
3350 (native mode)	115

However, the maximum size of VSAM data sets used by CMS, by device type, is:

Device	Maximum Number of Cylinders per VSAM Minidisk
2314/2319	200
3330, Models 1 and 11	404
3340-35	348
3340-70	696

For VSAM data sets in CMS, the 3350 is not supported in native mode and the 3330 Model 11 is supported only as a virtual 3330 Model 1. These limitations exist because the CMS/VSAM support is based on DOS/VS VSAM.

Figure 10 shows a virtual machine configured to execute CMS. A minimum CMS configuration would not include the virtual tapes.

Figure 11 is 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; the system operator attaches the tapes to the virtual machine when they are needed.

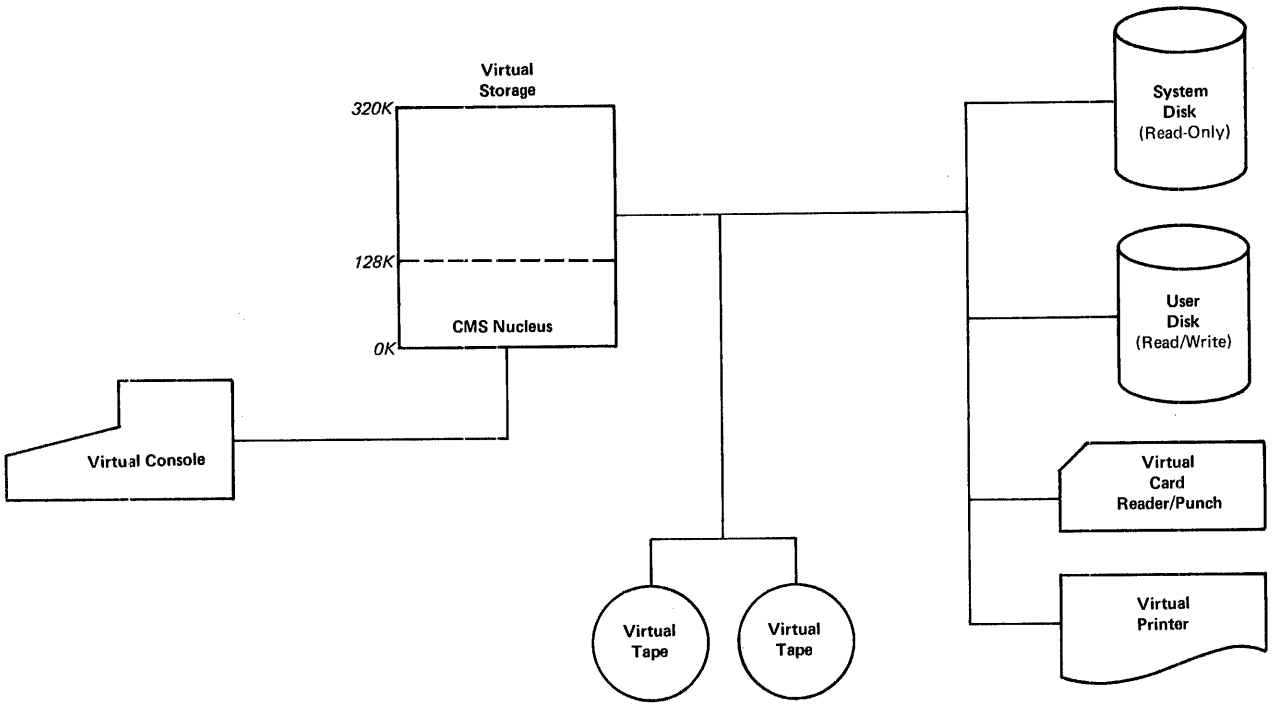

Figure 10. Sample CMS Configuration

The first virtual disk shown in Figure 11 is defined by the LINK control statement 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 is defined in Figure 11 by the MDISK control statement and is owned by the user SMITH. It has virtual address 191, is a minidisk located on the real volume labeled CMSVL1, and occupies five 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. VM/370 Directory Entry for a CMS Machine

CMS File System

CMS supports 2314, 2319, 3330, 3333, 3340, and 3350 disks. The CMS FORMAT command formats the tracks of a CMS disk into 800-byte blocks. The CMS file system manages these 800-byte blocks so that the user appears to have logical fixed- or variable-length records, and sequential or direct access to files.

The CMS support of VSAM provides CMS users with read, update, and write access to OS and DOS VSAM data sets on minidisks and real disks. CMS file size, record length, and minidisk size restrictions do not apply to VSAM data sets used in CMS. Also, the IBCDASDI program (or the OS/VS and DOS/VS disk initialization programs), and not the CMS FORMAT command, must be used to format VSAM data sets used in CMS. The OS/VS and DOS/VS disk initialization programs should be used only for full disks. The IBCDASDI program must be used for minidisks.

CMS has files that contain macro libraries and program libraries, and commands to use and update these libraries. 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 can 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 and the letter defines 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 under "Virtual Machine I/O Management" in the "Control Program" 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 the disk directory that has an entry for the file. The numeric part of the filemode defines the type of file access permitted. It indicates whether the file is private, read/write, read-only, or in simulated OS format. The filemode, in conjunction with the read/write status of the disk, determines which files on a disk can be accessed. 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 other user that shares that disk, and has 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 in 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, 3340, or 3350 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 Master File Directory, which contains format and size information for each file on the virtual disk and includes a pointer to the file's chain link records. (The CMS Master File Directory, or a specified subset of it, called the User File Directory, 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 is changed.) Each file's chain link records point to the 800-byte records of that file.

Figure 12 illustrates the CMS file structure. The User 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.

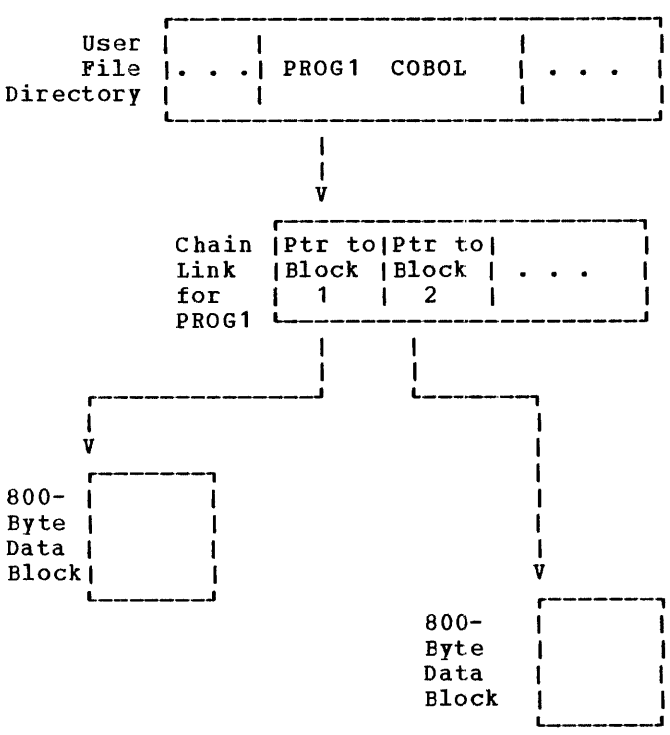

Figure 12. CMS File Format

CMS automatically opens and closes all accessed files (including spool files) for each command or user program it executes. Files can be spooled between virtual machines to transfer files between users. Service programs, which can be invoked by CMS commands, manipulate files; they can write an entire disk or specific disk file 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 commands. Tape labels are not supported by CMS.

When a compiler is invoked, CMS dynamically allocates compiler work files on whichever active user disk has the most available space (the location of these work files may also 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 or VSAM DOS files and sequential, partitioned, or VSAM 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 write or update OS data sets or DOS files. However, in CMS the user can read, write, or update VSAM data sets.

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

Initialization and Dump Restore

The OS IBCDASDI 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.

The CMS FORMAT command initializes minidisks for CMS. However, the IBCDASDI program must be used to initialize any minidisk that is used with VSAM files or catalogs.

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

The DASD Dump Restore (DDR) program of VM/370, which executes standalone or under CMS, dumps, restores, and displays 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, "a" can represent "assemble"). CMS keeps 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. To execute a program, the user must enter 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: CMS User's Guide.

Program Development and Execution

CMS has a wide range of programming development capabilities, it can:

- Create and compile source programs
- Build test files
- Execute and test programs
- Debug programs at the terminal

CMS commands are especially useful for OS and DOS/VS program development.

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; it must either be defined by his directory entry or added to his virtual machine configuration via 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." One exception is that disks that are to contain VSAM data sets must first be initialized with the IBCDASDI program (the DOS/VS or OS/VS disk initialization program may be used for full packs), instead of with the CMS FORMAT command.

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. However, a DLBL command must be issued for OS VSAM files used under CMS. The ASSGN and DLBL commands define devices and DASD files that are used in the CMS DOS environment.

Except for the CMS commands that specifically support VSAM, no other CMS commands (such as TYPE, PRINT, or EDIT) can access DOS or OS VSAM data sets. The CMS support for Access Method Services and VSAM is based on DOS/VS; however, the ISAM Interface Program is not supported.

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 part of the fileid, such as the filetype. The ERASE command deletes a file or a group of related files from a user's read/write disk.

The CMS Editor

The CMS Editor consists of the EDIT command and its subcommands. 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 editor then displays the record containing this data 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 previously issued the EDIT subcommand VERIFY ON, the editor also displays the changed line 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 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, is in the VM/370: CMS User's Guide.

A text processor called SCRIPT/370 is available as an IUP (Installed User Program) for use under CMS. (See "Appendix B: 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 CMS commands that invoke OS compilers and the assembler are: ASSEMBLE, ASM3705, VSBASIC, COBOL, FORTGI, FORTHX, GOFORT, PLIOPT, and PLIC. On each of these command lines, the user can specify CMS options, and also language processor options, that are identical to those coded on an OS EXEC card when the language processor is invoked from OS.

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.

The CMS commands that invoke DOS compilers are: FCOBOL and DOSPLI. The COBOL user specifies the compiler options on an OPTION command which precedes the FCOBOL command. The PL/I user must specify compiler options on an *PROCESS statement which is placed in front of the PL/I source program.

Under CMS the user can execute programs that read DOS sequential files and those that read and write DOS VSAM files. However, under CMS, the user cannot execute DOS programs that use direct or indexed DOS files.

Also, DOS private and system source statement, relocatable, and core image libraries can be read in CMS. Several CMS commands provide the library services similar to those found in DOS/VS. The CMS commands are SSERV, RSERV, PSERV, ESERV, and DSERV. DOS private libraries that are to be read in CMS must first be identified with the CMS ASSGN and DLBL commands.

The DOS linkage editor is simulated in CMS. Files to be link-edited can be read from DOS libraries. The CMS DOSLKED command link-edits files and places the output in a CMS file called DOSLIB. The CMS DOSLKED command accepts the DOS linkage editor control statements (ACTION, PHASE, INCLUDE, and ENTRY) as input.

The user executes DOS programs under CMS using the FETCH and START commands. Libraries to be searched must first be specified on a CMS GLOBAL command. Although the user can sometimes create CMS module files from DOS programs by using the FETCH and GENMOD commands, this procedure is not recommended.

CONTROL COMMANDS

The CMS user is able to define certain system functions with the SET command. The functions include: the amount of information 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 functions.

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 are 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. VS APL, 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 various PL/I, FORTRAN, and COBOL compilers. VS BASIC and VS APL also execute under CMS and include interactive debug facilities as part of the Program Product. For information on programming languages and compilers that can be used with CMS, see "Appendix B: 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.

The VS APL interpreter executes in the conversational environment of CMS. The interpreter responds interactively to input keyed in at the terminal, displaying results when called for or displaying diagnostic messages when input is invalid.

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 compilers and the VS APL interpreter, CMS simulates the execution of many of the OS and DOS macros. For OS, the sequential, direct, and partitioned access methods are logically simulated. CMS keeps these data records in the chained 800-byte blocks, which are standard to CMS, and simulates the OS data set characteristics internally. CMS supports OS VSAM data sets. Many OS Supervisor Call functions including GETMAIN/FREEMAIN and TIME are simulated.

For DOS, the sequential access method and virtual storage access method are supported as well as DOS/VS library access. Note that sequential output files are written as CMS files and that DOS/VS libraries can only be read.

The OS and DOS macros that are not simulated include those that support the Indexed Sequential Access Method (ISAM) 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 execute under CMS while COBOL programs that use ISAM do not.

Functions related to multitasking are either ignored by CMS or modified to achieve single task execution. The DOS/VS Sort/Merge program is not supported under CMS. See the VM/370: System Programmer's Guide for more information about CMS's OS and DOS macro support.

ALTERNATING OPERATING SYSTEMS

If a program to be tested uses OS and DOS functions that are not simulated, or if the program is designed for some other operating system, 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 and begin the compilation. When the job stream completes, the user must reload CMS 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 same procedure.

With this technique, a user can compile OS ISAM programs under CMS and then test them under OS. See the ISAM and dynamically modified channel program

restrictions listed 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, to COBOL programmers using ANS Version 4 COBOL in the OS COBOL Interactive Debug Program Product, and to PL/I programmers using the PL/I checkout compiler. VS BASIC and VS APL include interactive debug facilities as part of the Program Product.

CMS BATCH FACILITY

The CMS Batch Facility is a VM/370 programming facility that executes under CMS. It allows a VM/370 user to execute jobs in batch mode by sending jobs from either his own virtal 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 sent to the batch machine's virtual card reader from either the 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, or large programs. Thus, interactive users can continue working at their terminals while their time-consuming jobs are executed 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 executed under CMS, apply to programs executed 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 VM/370: CMS User's Guide.

Remote Spooling Communications Subsystem

The VM/370 Remote Spooling Communications Subsystem (RSCS) is the VM/370 component that transfers 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.

An RSCS virtual machine controls the transferring of files.

RSCS has a supervisor and line drivers. The supervisor is an interface between the CP spool system and the RSCS line driver. The line drivers 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 Teleprocessing Network

The RSCS network consists of a real CPU, transmission control units, and BSC telecommunications lines and remote stations.

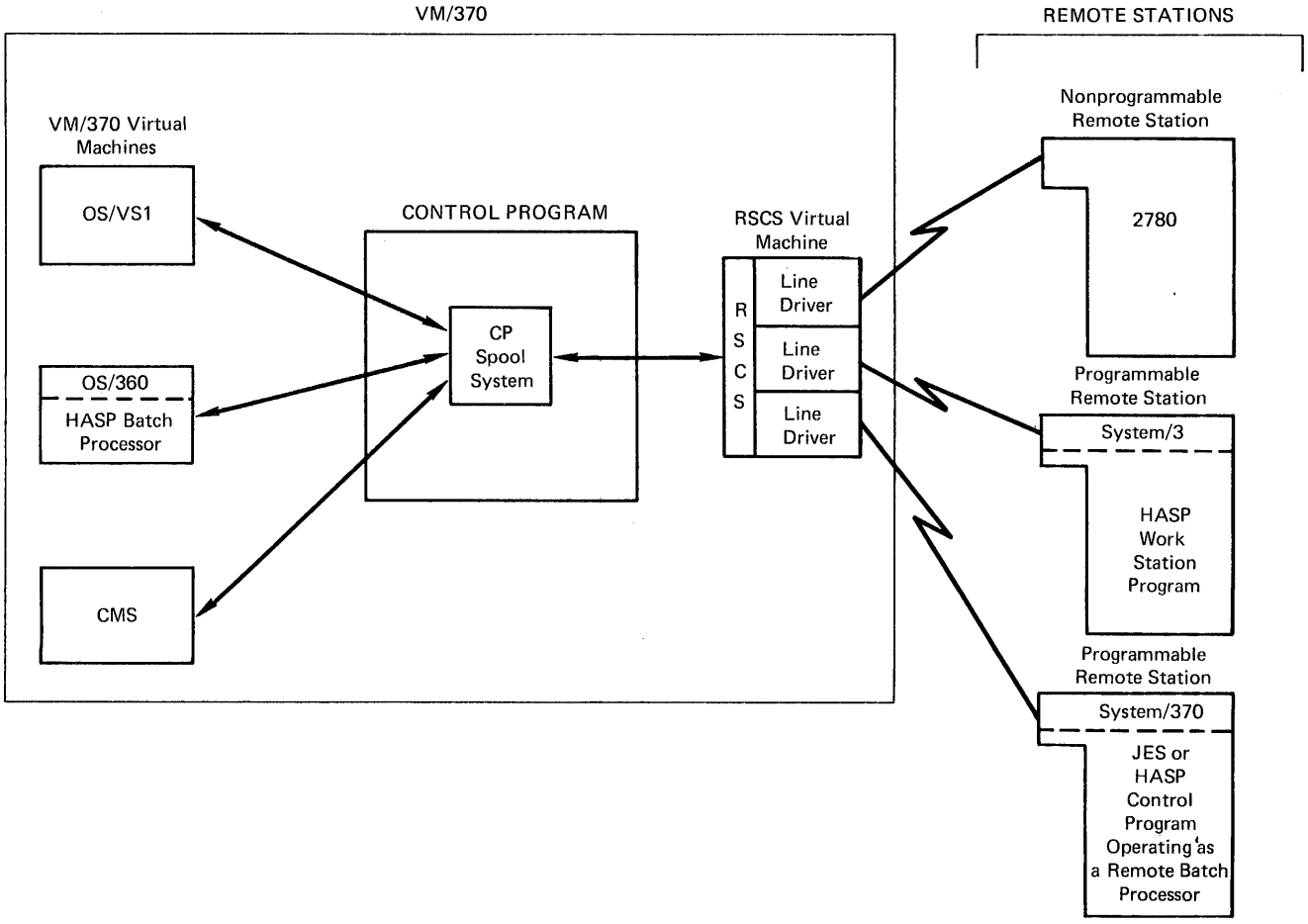

Figure 13. A VM/370 RSCS Teleprocessing Network

THE REAL CPU

A real CPU that is executing VM/370, and RSCS in a virtual machine, is the center of the RSCS teleprocessing network. The operator of the RSCS virtual machine controls the network by issuing RSCS commands from his terminal.

The CP spool system is an integral part 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

Teleprocessing equipment (transmission control units, data sets, and communication lines) is required to control the teleprocessing network. Transmission control units control the transmission of data between the real CPU 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 and nonprogrammable.

Programmable Remote Stations

Programmable remote stations are I/O configurations that include a computer, such as a System/3, System/32, 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, if the programmable remote station does not have a HASP- or ASP-type of batch processor, 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 the VM/370: Remote Spooling Communications Subsystem (RSCS) User's Guide.

Using RSCS

The facilities of RSCS are selected and controlled by 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.

To link a remote station to a virtual machine, the RSCS operator issues the START command. Then, to begin transmitting to the RSCS virtual machine, the remote station operator transmits a SIGNON card.

Once the link between a remote station and the RSCS virtual machine is established via START and SIGNON, files can be transferred to and from that location. An ID card, which specifies the eventual destination, must precede each file transmitted from the remote station to the RSCS virtual machine. The RSCS virtual machine uses the information on the ID card to transmit the file to the designated virtual machine. If tag information is also supplied, RSCS can transmit the file to another remote station.

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

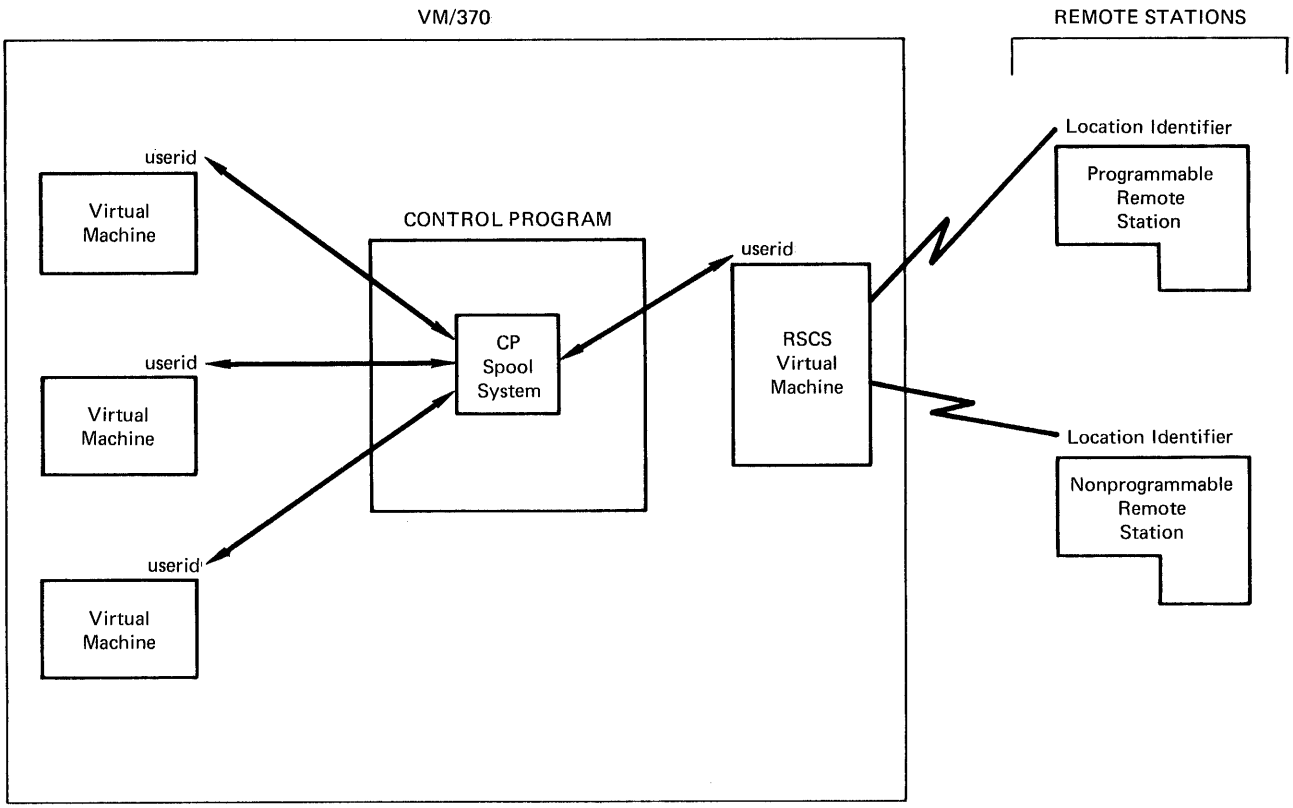

Figure 14. Linking Virtual Machines and Remote Stations

COMMANDS USED TO TRANSMIT FILES AND TO CONTROL THE RSCS NETWORK

File transmission 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 issue the CP TAG command to name the location identifier of the destination that is to receive the file. The CP SPOOL command and a command that closes the file being transmitted, such as the CMS PUNCH or PRINT command, cause the file to be sent to the RSCS virtual machine for processing. The RSCS virtual machine then transmits the file across its network. The CP TAG and SPOOL commands control the transmission of files from virtual machines to remote stations, whereas the ID card controls the transmission of files from remote stations to virtual machines.

Commands for Controlling the RSCS Virtual Machine and Remote Stations

RSCS has commands and control cards that control 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, functions such 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

If a batch system has a remote job entry capability, such as HASP and JES2, it requires its own control statement in order to execute correctly. When RSCS is operating as a remote job entry system for one of the batch processors, the RSCS operator can transmit to that batch processor via 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: 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 135-3
IBM System/370 Model 138
IBM System/370 Model 145
IBM System/370 Model 145-3
IBM System/370 Model 148
IBM System/370 Model 155 II
IBM System/370 Model 158
IBM System/370 Model 158 MP (uniprocessor mode only)
IBM System/370 Model 165 II
IBM System/370 Model 168
IBM System/370 Model 168 MP (uniprocessor mode only)
IBM System/370 Model 168 Attached Processor System (uniprocessor mode only)

These machines must be equipped with the Dynamic Address Translation facility and at least 245,760 bytes of real storage. They must operate in extended control mode. The Models 135 and 145 require the System Timing Facility and 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.

For information about storage requirements, refer to the VM/370: Planning and System Generation Guide.

SYSTEM CONSOLES

The following system consoles and terminals are supported by VM/370 as virtual consoles (simulated as 3215 consoles):

- IBM 2150 Console with 1052 Printer-Keyboard, Model 7
- IBM 3066 Models 1 and 2 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 Consoles for the System/370 Models 135-3, 138, 145-3, and 148 in printer-keyboard mode (with the 3286 printer, Model 2 required) or in display mode
- 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 3275 Display Station, Model 2, with integrated control unit (remote attachment)
- 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 printers can be attached to local or remote 3270's:

- IBM 3284 Printer, Model 2 (local or remote attachment) or Model 3 (remote attachment only)
- IBM 3286 Printer, Model 2 (local or remote attachment) or Model 3 (remote attachment only)
- IBM 3288 Line Printer, Model 2 (local or remote attachment)

Terminal Control Units

The supported terminal control units are:

- IBM 3271 Control Unit, Model 2, for remote attachment of 3277 display stations, 3284 Printers, 3286 Printers, and 3288 Line Printers.
- IBM 3272 Control Unit, Model 2, for local attachment of 3277 display stations, 3284 Printers, and 3286 Printers.

Transmission Control Units

The supported transmission control units are:

- 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.

The devices supported for nonprogrammable remote stations are:

- 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, (operating in 2770 compatibility mode)
- IBM 3780 Data Communications Terminal

The transmission control units supported for programmable remote stations are:

- ICA (Integrated Communications Adapter)
- IBM 2701 Data Adapter Unit with Synchronous Data Adapter, Type II
- IBM 2703 Transmission Control Unit with Synchronous Terminal Control (except for 3770)
- IBM 3704 Communications Controller (in EP mode only)
- IBM 3705 Communications Controller (in EP mode only)

The systems supported for remote job entry into RSCS are:

- IBM System/360 Models 20, 22, 25, 30, 40, 50, 65, 75, 85, 195
- IBM System 370 Models 115, 125, 135, 135-3, 138, 145, 145-3, 148, 155, 155 II, 158, 165, 165 II, and 168
- IBM 1130 System
- IBM System 3 Models 6, 8, 10, 12, and 15
- IBM System/32
- 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.

The software subsystems supported for RSCS are:

- 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

VM/370 supports the following 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, and the IBM 3344 Direct Access Storage, Model B2.
- IBM 3350 Direct Access Storage, Models A2 and B2.

DIRECT ACCESS CONTROL UNITS

VM/370 supports the following control units for DASD:

- 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 3340 Direct Access Storage Facility, Model A2.
- IBM 3345 Storage and Control Frame, Models 3, 4, and 5 on the Model 145 (with the standard ISC) for 3330 Disk Storage Models 1 and 2, 3333 Disk Storage and Control, Models 1 and 11, 3340 MOdel A2, 3344, and 3350
- 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, 3340 Model A2, 3344, and 3350
- 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, 3340 Model A2, and 3344 Model B2
- IBM ISC (Integrated Storage Control) (#4660) on the Model 145 for 3330 Models 1 and 2, 3333 Models 1 and 11, 3340 Model A2, 3344, and 3350.
- 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, 3340 Model A2, 3344, and 3350

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

VM/370 supports the following 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 (9-track only)
- IBM 3411 Magnetic Tape Unit and Control, Models 1, 2, and 3 (9-track only)
- IBM 3420 Magnetic Tape Unit, Models 3, 4, 5, 6, 7, and 8

MAGNETIC TAPE CONTROL UNITS

VM/370 supports the following 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

VM/370 supports the following printers:

- IBM 1403 Printer, Models 2, 3, 7 and N1
- IBM 1443 Printer, Model N1
- IBM 3203 Printer, Model 4 (System/370 Models 138 and 148 only)
- IBM 3211 Printer

READER/PUNCHES

VM/370 supports the following readers and 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

VM/370 supports:

- 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	9-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

Most of VM/370 is written in VM/370 Assembler language, using the instructions available only on an IBM System/370 with dynamic address translation.

Support Considerations

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

Appendix B: Language Processors and Emulators

VM/370 Assembler

The 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) and the Remote Spooling Communications Subsystem (RSCS) are components of VM/370 and are 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. 5796-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
DOS PL/I Optimizing Compiler	5736-PL1
DOS PL/I Resident Library	5736-LM4
DOS PL/I Transient Library	5736-LM5
DOS PL/I Optimizing Compiler and Libraries	5736-PL3
DOS/VS COBOL Compiler & Library	5746-CB1
DOS/VS COBOL Object Library	5746-LM4
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 & 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
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
VS BASIC Processor	5748-XX1
VS APL	5748-AP1

Figure 15. IBM Program Products

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

MUSIC supports several programming languages, for example: OS FORTRAN G1, VS BASIC, OS ANS COBOL, APL and OS Assembler F. MUSIC also has a batch facility, Context Editor, accounting, sort programs, and command language. For additional information about MUSIC, see the MUSIC: General Information Manual, Order No. G320-1238.

The VS/REPACK IUP package, IBM Program No. 5796-PDZ, allows the user to collect and graphically display the pattern of activity of his program or system, analyze this data to predict how rearranging portions of the program will improve locality of reference (thereby reducing its working set size and paging activity), and verify that the program is operating correctly.

A data reduction program is available as an IUP: the statistics-generating package for VM/370 (VM/SGP), IBM Program No. 5796-PDD. VM/SGP reduces the data that is collected by the VM/370 measurement facility, thus providing useful information for installation management, systems programmers, consultants, and users.

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 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).

System/370 Model	S/360 Model 20	1401 1440 1460	1401 1440 1460 1410 7010	7070 7074	7080	709 7090 7094 7094II
135, 135-3, 138	#7520	#4457				
145, 145-3, 148		#4457	#4458			
155 II, 158			#3950	#7117		
165 II				#7117	#7118	#7119
168				#7127	#7128	#7129

Figure 16. Integrated Emulators that Execute under VM/370

Appendix C: VM/370-Related Publications for CMS Users

This appendix lists 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
VM/370: Commands (General User)[1]	GX20-1961
VM/370: Commands (Other than General User)[1]	GX20-1995
VM/370: CMS Command and Macro Reference	GC20-1818
VM/370: CMS User's Guide	GC20-1819

Corequisite Publications

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

Also Available

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

The publications that are relevant to a particular type of CMS user are listed by categories of CMS users. 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/370 Bibliography, Order No. GC20-0001.

Note: In some cases, the titles are 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 three reference summaries are available separately or can be ordered at the same time by using Order No. GBOF3576.

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 following list.

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
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 Compiler and Library Installation Reference Material	SC28-6481
OS/VS COBOL Compiler and Library Programmer's Guide	SC28-6483
DOS/VS COBOL Compiler and Library Programmer's Guide	SC28-6478
DOS/VS COBOL Compiler and Library Installation Reference Material	SC28-6479
VM/370 CMS User's Guide for COBOL	SC28-6469

DOS/VS VSAM and CMS VSAM Users

DOS/VS Data Management Guide	GC33-5372
DOS/VS Supervisor and I/O Macros	GC33-5373
DOS/VS Access Method Services User's Guide	GC33-5382

OS/VS VSAM User

OS/VS VSAM System Information	GC26-3835
OS/VS2 Programming Library: Data Management System	GC26-3830
OS/VS VSAM Programmer's Guide	GC20-3818
OS/VS Access Method Service	GC35-0009
OS/VS VSAM Planning Guide	GC26-3799
OS/VS VSAM Options for Advanced Applications	GC26-3819
OS/VS Data Management Services Guide	GC26-3783
OS/VS Access Method Services	GC26-3836
OS/VS Planning and Use Guide	GC24-5090
OS/VS2 Access Method Services	GC26-3841
Planning for Enhanced VSAM under OS/VS	GC26-3842

PL/I User

OS PL/I Optimizing Compiler, Program Product Specifications	GC33-0022
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 PL/I Optimizing Compiler CMS User's Guide	SC33-0037
OS TSO PL/I Optimizing Compiler	SC33-0029
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/I 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	SC33-0047
OS PL/I Resident Library Program Product Specifications	GC33-0023

OS PL/I Transient Library Program Product Specifications	GC33-0024
DOS PL/I Checkout Compiler Program Product Specifications	GC33-0016
DOS PL/I Optimizing Compiler Installation	SC33-0020
DOS PL/I Optimizing Compiler Programmer's Guide	GC33-0008
DOS PL/I Optimizing Compiler CMS User's Guide	SC33-0051
DOS PL/I Optimizing Compiler Messages	SC33-0021
DOS PL/I Resident Library Program Product Specifications	GC33-0017
DCS PL/I Transient Library Program Product Specifications	GC33-0018

VS APL User

VS APL: Terminal User's Guide for CMS	SC20-9067
VS APL for CMS: Writing Auxiliary processors	SH20-9068
VS APL: Reference Summary	SX26-3712
VS APL: Installation Reference Material	SH20-9065
VS APL: General Introduction	GH20-9064
APL Language: Concepts and Reference	GC26-3853

Index

YOUR COMMENTS PLEASE . . .

Your views about this publication may help improve its usefulness; this form will be sent to the author's department for appropriate action. Using this form to request system assistance and/or additional publications or to suggest programming changes will delay response, however. *For more direct handling of such request, please contact your IBM representative or the IBM Branch Office serving your locality.* Your comments will be carefully reviewed by the person or persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Fold Fold

First Class
Permit 81
Poughkeepsie
New York

Business Reply Mail
No postage stamp necessary if mailed in the U.S.A.

Postage will be paid by:

International Business Machines Corporation
Department D58, Building 706-2
PO Box 390
Poughkeepsie, New York 12602

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)

Cut or Fold Along Line

IBM Virtual Machine Facility/370: Introduction Printed in U.S.A. GC20-1800-6

READER'S COMMENTS

Title: IBM Virtual Machine Facility/370: Introduction

Order No. GC20-1800-6

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.

Trim Along This Line