File No. S370-20
Order No. GC20-1800-9

Systems

IBM Virtual Machine Facility/370: Introduction

Release 6 PLC 4

This publication introduces VM/370, and 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 a system control program (SCP) that tailors the resources and capabilities of a single System/370 computer to provide concurrent users their one unique (virtual) machine. 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 a Interactive Problem Control System (IPCS), which provides problem analysis and management facilities.

The first section of the publication is an introduction; it describes what VM/370 can do. The second, third, fourth, and fifth sections describe the Control Program, Conversational Monitor System, Remote Spooling Communications Subsystem, and Interactive Problem Control System respectively. The appendixes include information about VM/370 publication-to-audience relationship and VM/370-related publications for CMS users.

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

Tenth Edition (March 1979)

This edition (GC20-1800-9) together with Technical Newsletter GN25-0489, dated August 1, 1979, applies to Release 6 PLC 4 (Program Level Change) of IBM Virtual Machine Facility/370 and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters.

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

Changes are periodically made to the information 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.

It is possible that this material may contain references to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Publications are not stocked at the address given below; 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. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

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), the Remote Spooling Communications Subsystem (RSCS), and the Interactive Problem Control System (IPCS).

This publication contains five sections and two 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.
- "Interactive Problem Control System" describes the organization and function of IPCS.
- "Appendix A: VM/370 Publication-to-Audience Relationship"
- "Appendix B: 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, GR20-4260.

RELATED PUBLICATIONS

IBM Virtual Machine Facility/370:

CMS Command and Macro Reference, GC20-1818

CMS User's Guide, GC20-1819

Commands (General User), GX20-1961[1]

Commands (Other than General User), GC20-1995[1]

CP Command Reference for General Users, GC20-1820

Glossary and Master Index, GC20-1813

Operating Systems in a Virtual Machine, GC20-1821

Operator's Guide, GC20-1806

Planning and System Generation Guide, GC20-1801

Quick Guide for Users, GX20-1926[1]

Release 6 Guide, GC20-1834

Remote Spooling Communications Subsystem (RSCS) User's Guide, GC20-1816

System Messages, GC20-1808

System Programmer's Guide, GC20-1807

Features Supplement, GC20-1757

Terminal User's Guide, GC20-1810

Interactive Problem Control System (IPCS) User's Guide, GC20-1823

Data Areas and Control Block Logic, SY20-0884

System Logic and Problem Determination Guide Volume 1 Control Program (CP), SY20-0886[2]

System Logic and Problem Determination Guide Volume 2 Conversational Monitor System (CMS), SY20-0887[2]

System Logic and Problem Determination Guide Volume 3 Remote Spooling Communications Subsystem (RSCS), SY20-0888[2]

---

[1] These three publications are available as a unit, GBOF-3576

[2] These three volumes are available as a unit, SBOF-3802

In order to use the CPEREP command, the following publications are required:

IBM Virtual Machine Facility/370: OLTSEP and Error Recording Guide, GC20-1809

OS/VS, DOS/VSE, VM/370 Environmental Recording, Editing, and Printing (EREP) Program, GC28-0772

The first publication provides general information on usage and detailed information on command operands applicable only to VM/370. The second publication provides detailed information on the operands that are common to OS/VS, DOS/VSE, and VM/370.

Program logic information describing the interface between CMS and OS/VS EREP, and describing OS/VS EREP, is contained in the following:

IBM Virtual Machine Facility/370: Service Routines Program Logic, SY20-0882

OS/VS, DOS/VSE, VM/370 Environmental Recording, Editing, and Printing (EREP) Program Logic, SY28-0773

In addition, all EREP messages are contained in the OS/VS, DOS/VSE, VM/370 Environmental Recording, Editing, and Printing (EREP) Messages, GC38-1045.

MISCELLANEOUS

A complete description of VM/370 restrictions is available in VM/370 System Messages and VM/370 Planning and System Generation.

Figure 1 is an overview of the VM/370 library, and supplemental support, 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.

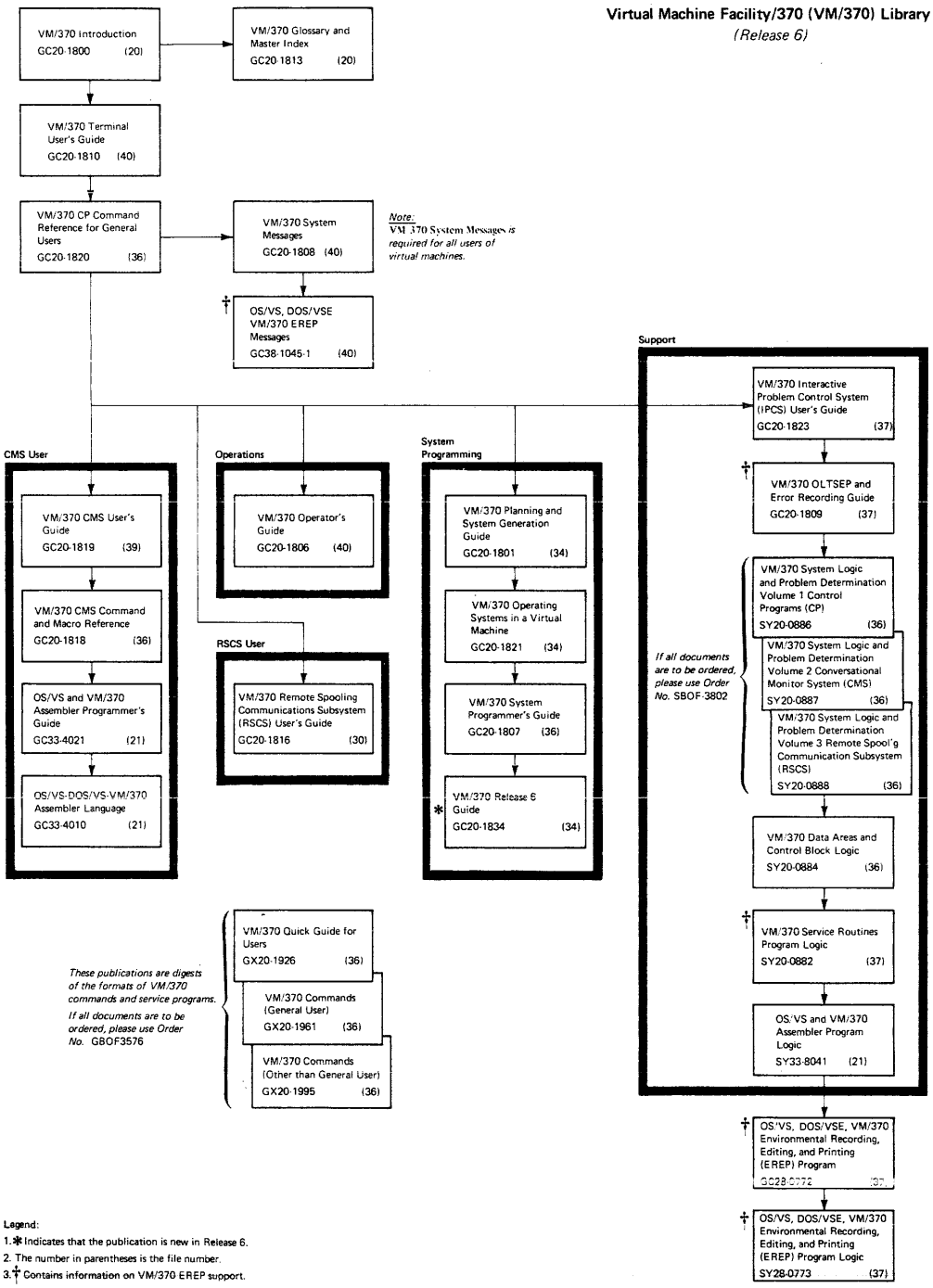


Figure 1. Virtual Machine Facility/370 Library

Contents

The entries in this Table of Contents are accumulative. They list additions to this publication by the following VM/370 System Control Program Product:

- VM/370 Basic System Extensions, Program Number 5748-XX8
- VM/370 System Extensions, Program Number 5748-XE1

However, the text within the publication is not accumulative; it only relates to the one SCP program product that is installed on your system. Therefore, there may be topics and references in this Table of contents that are not contained in the body of this publication.

Summary of Amendments
for GC20-1800-9
As Updated by GN25-0489
VM/370 Release 6 PLC 4

Channel-set Switching Support

New: Program and Documentation

For systems equipped with Channel-set Switching, function has been added to VM/370 that allows the operator to switch channels from one processor of an attached processor system to the other processor. The operator can invoke channel switching via CP's VARY PROCESSOR command or it can occur automatically upon detection of a severe hardware malfunction on the main (I/O) processor. After the switching operation, VM/370 continues operation in uniprocessor mode. This facility enhances the reliability, availability and serviceability of VM/370. For details, refer to the topic: "VM/370 Attached Processor Support."

Summary of Amendments
for GC20-1800-9
VM/370 Release 6 PLC 1

3800 PRINTING SUBSYSTEM

New: Program and Documentation

VM/370 supports the 3800 Printing Subsystem as a high-speed printing device that uses electrophotographic and laser technology.

VM/370 allows virtual machines to attach the 3800 printer (with full capabilities) as a dedicated device. In addition, VM/370 supports the 3800 Printing Subsystem as a real spooling device.

3850 MASS STORAGE SYSTEM

New: Program and Documentation

VM/370 supports user minidisks on Mass Storage System (MSS) 3330V volumes, where the minidisk appears to the virtual machine to be defined on a permanently mounted IBM 3330-1 drive.

MSS volumes may also be dedicated and attached to a virtual machine as either a 3330V or a 3330-1 device.

LOGON/AUTOLOG/LINK JOURNALING

Changed: Program and Documentation

VM/370 now supports a function that will aid in the prevention of unauthorized use of virtual machines and minidisks.

When the new option (JOURNAL operand) is specified, passwords cannot be entered on the same line as the LOGON/AUTOLOG or LINK commands. All unsuccessful attempts to do so will be recorded, when the threshold has been reached, the system administrator is notified.

SHARED SEGMENTS

New and Changed: Program and Documentation

A new option has been added that modifies the protection of shared segments.

If the PROTECT option is specified for a shared system, the user who alters a shared page in that system receives a message indicating the named system containing the shared page and the hexadecimal location. VM/370 places the user in console function mode and returns the modified page(s) to free storage. The next reference to the page(s) causes a fresh copy to be paged from storage.

VM/370 provides the option of running without protection against changes to shared segments. If a shared segment is specified as unprotected, any changes will be seen by all other users of that shared segment.

REMOTE SPOOLING COMMUNICATIONS SUBSYSTEM

VM/370 provides a new program product (Remote Spooling Subsystem Networking) that enhances the networking facilities of the VM/370 system, and allows a VM/370 system to use the Network Job Interface/Network Job Entry (NJI/NJE) protocol.

MISCELLANEOUS

Changed: Documentation only

Literary and editorial changes have been made throughout this publication.

The entire "Introduction" to this publication has been rewritten and should be read in its entirety.

Summary of Amendments
for GC20-1800-8
as updated by GN25-0454
VM/370 Release 5 PLC 12

VARY PROCESSOR ONLINE/OFFLINE SUPPORT

New and Changed: Programming support

VM/370 attached processor (AP) provides support to enhance the reliability, availability, and serviceability of the 158 and 168 attached processors, and the 158 and 168 asymmetric MP systems.

This support allows the attached processor to be taken offline, make needed repairs, and then bring it back online without affecting the main processor.

Introduction

Virtual Machine Facility/370 (VM/370) is a system control program. It manages the resources of an IBM System/370 in such a way that multiple users have a functional simulation of a computing system (a virtual machine) at their disposal.

That is, the virtual machine runs as if it were a real machine simulating both hardware and software resources of the system. These simulated resources can be either shared with other virtual machines or alternately allocated to each machine for a specified time. Furthermore, virtual machines can run the same or different operating systems simultaneously. Thus, the individual user can create and adapt his virtual machine to meet his own special needs.

VM/370 Components

The VM/370 system consists of four components; the control program (CP), conversational monitor system (CMS), remote spooling communications subsystem (RSCS), and interactive problem control system (IPCS). Each of these components control its unique part of the system. Together, these components provide the virtual machine with time sharing, remote spooling, and problem reporting for System/370 uniprocessor and attached processor systems.

CONTROL PROGRAM

The control program (CP) executes in a real machine controlling the resources of that machine. CP is the vehicle that is used to create concurrent virtual machines.

CONVERSATIONAL MONITOR SYSTEM

The conversational monitor system (CMS) is a single-user operating system (designed to operate in VM/370) that can execute in one or more virtual machines. CMS provides a wide range of general-purpose, conversational time sharing functions.

REMOTE SPOOLING COMMUNICATIONS SUBSYSTEM

The remote spooling communication subsystem (RSCS) is a single-user operating system that runs under CP. RSCS executes in one or more virtual machines and transfers data between virtual machines and remote users.

INTERACTIVE PROBLEM CONTROL SYSTEM

The interactive problem control system (IPCS) is a group of commands and controls that execute under CMS to provide problem analysis and management facilities. IPCS standardizes the process of reporting problems and includes a method for identifying duplicate problems within the system. It also provides the user with the capability of viewing and diagnosing CP abend dumps through the virtual operator's console.

The Virtual Machine

A virtual machine is functionally equivalent to a real system. It has simulated hardware and software resources that operate in a real computer under CP. Each virtual machine is defined in the VM/370 directory; the directory describes its simulated storage, I/O devices, and console.

Figure 2. shows three virtual machines executing concurrently under CP on an IBM System/370 model 138, with 1024K of real storage. One machine is doing production work under the current release of DOS and the other two virtual machines are executing CMS: one virtual CMS machine is available for each of two separate conversational users.

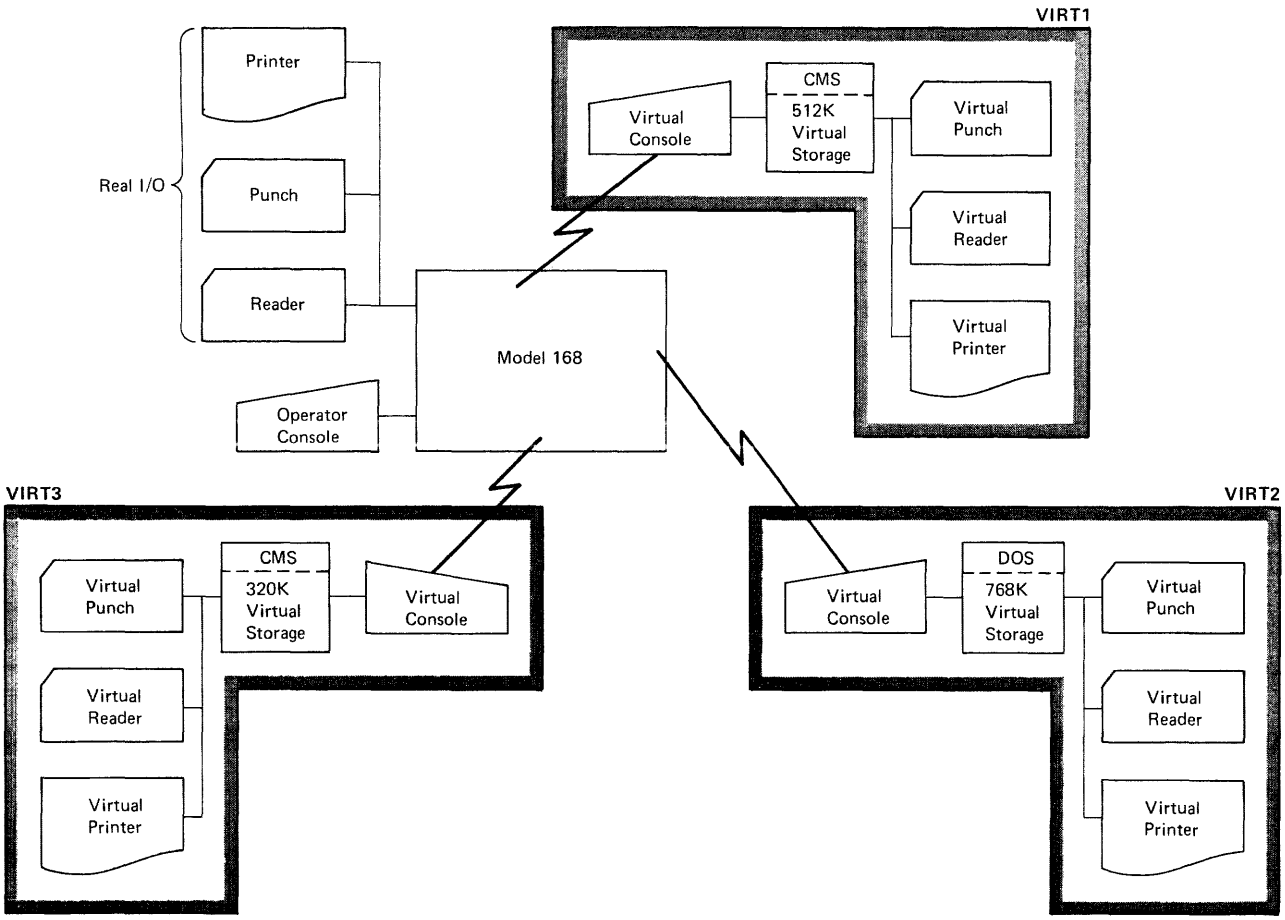


Figure 2. Simultaneous Operations of Virtual Machines

Multiple virtual machines permit several systems to execute concurrently. Thus, production work may be run in one or more virtual machines while other virtual machines are executing:

- A terminal-oriented conversational system
- A remote spooling subsystem, which transmits bulk data to and from remote locations
- A back-release system that executes application programs below the current service level of the production system
- A current-release system with new program temporary fixes (PTFs) applied
- A new release or option that needs to be generated and tested
- A conversion test from one operating system to another, such as from DOS/VSE to OS/VS1

VIRTUAL STORAGE

A virtual storage system can simulate real storage within a range of from 8K (8192) bytes to 16 megabytes (the maximum virtual storage size). The virtual storage for each virtual machine is defined in the VM/370 directory.

Virtual storage extends beyond the size of real storage and is not limited by the amount of real storage. It is possible for the combined virtual storage of several virtual machines to be greater than the real storage.

Virtual storage can be managed and protected through segmentation. Each segment is 64K and there are from 1 to 256 segments depending on the size of virtual storage.

As a storage protection feature for all virtual machines, page and segment tables are accessible only to VM/370. One virtual machine operating system can not access or alter the virtual storage of another virtual machine.

Refer to the "Control Program" section of this publication for more detailed information.

VIRTUAL PROCESSOR

CP provides each virtual machine a single virtual processor to execute instructions and receive interruptions. In actuality, this virtual processor is the shared use of the real processor. CP simulates the privileged instructions, and the real processor executes the nonprivileged instructions.

VIRTUAL SYSTEM CONSOLE

In order to communicate with VM/370, there must be an interactive I/O device supported by the system. Any real terminal supported by VM/370 can become the virtual system console. (This can be very beneficial to the user with a limited system configuration.) The virtual system console is able to perform most of the functions of the real system console. Using the virtual system console a user can perform functions such as initializing an operating system, starting and stopping program execution, and displaying and changing the contents of registers and storage.

Virtual I/O Devices

The virtual machine supports the same devices as a real machine: It is the virtual machine, not VM/370, that controls them. The I/O configuration must be defined by the user in the VM/370 directory entries. However, additional I/O requirements can be met dynamically by the user via CP commands.

The user also has the option of assigning different address to his virtual devices, or using those of the real devices. In either case, CP converts the virtual address to its real counterpart and performs any necessary data translation. In addition, VM/370 allows the same virtual address to be used by multiple users.

VM/370 supports several types of virtual-to-real device mappings:

- Dedicated devices
- Virtual disks
- Virtual unit record devices
- Virtual transmission control units (virtual lines)
- Channel-to-channel adapters

DEDICATED DEVICES

Devices with a one-to-one correspondence between real and virtual are referred to as dedicated devices. Tape devices are always dedicated. The user can dedicate disks, terminals, and unit record devices to a virtual machine to improve the performance of that virtual machine. The 3800 high-speed printer is also supported as a dedicated device.

When a device is dedicated to a virtual machine, that device is completely controlled by the virtual machine. Dedicated devices can be assigned dynamically by CP commands, assigned at logon time by directory control statements, or established by dial-up connections to virtual 270x telecommunications control units or virtual lines. Dedicated devices are also useful when operating devices that, while available to the System/370, are not supported by VM/370.

Mass Storage System

The 3850 Mass Storage System (MSS) can be dedicated to as many as four virtual machines running on a VM/370 system. The mass storage control (MSC) in the 3851 Mass Storage Facility (MSF) accepts requests for data from one to four System/370 processors. Each of the connections can be attached either to a different processor or to a different VM/370 virtual machine. Other non-MSS virtual machines can run concurrently.

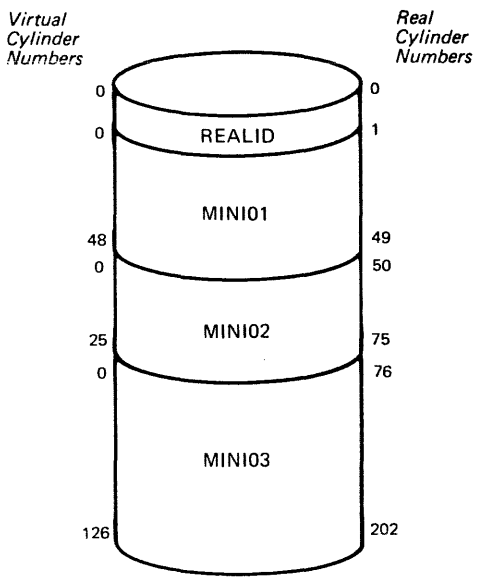
With MSS, there are available to the host processor as many as 192 DASD device addresses per DASD control unit. Up to 192

MSS volumes per control unit may be mounted and online at any given time. Each of these volumes may be accessed by a virtual machine for the same purposes that a real 3330-1 is used. Any virtual machine may have access to all or part of an MSS volume just as though it were a real 3330-1, with one exception: the VM/370 control program may not use MSS volumes for system residence, paging, or spooling.

When one or more of the MSCs channel ports is attached to the VM/370 system, the volume REALID in Figure 3 can be a MSS 3330V volume. The volume REALID will be dynamically mounted by the system whenever minidisk MINI01, MINI02, or MINI03 is required by a virtual machine.

VIRTUAL DISKS

Virtual disks are referred to as minidisks because more than one can be assigned to a full real disk. These disks vary in size and can be used by CMS, DOS, or OS. Figure 3 shows three virtual minidisks that are assigned to one real 2319 disk volume. These virtual disks can belong to one virtual machine or three different machines. The user can obtain, through VM/370 distribution, service programs that create and change minidisks.



Note: Each minidisk starts at virtual cylinder zero.

Figure 3. Real Disk Containing Minidisks

The minidisk owner can ensure the security of his disk by specifying a password option to his read/write disk or he can limit disk access to read-only. To avoid updating conflicts, VM/370 gives write access of a shared disk to one virtual machine at a time. However, the user can specifically request multiple access, if he so chooses.

VIRTUAL UNIT RECORD DEVICES

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. When setting his configuration, the user can include as many virtual unit record devices as he feels he will need. CP controls these virtual devices (as well as the real devices) and uses spooling to temporarily store input and virtual output data on direct access storage space. CP then transfers input data to the appropriate virtual machine reader and allows a real printer or punch to process output from the virtual device. Spooling is explained further under "Spooling Unit Record I/O" in the "Control Program" section of this publication.

VIRTUAL TRANSMISSION CONTROL UNITS (VIRTUAL LINES)

A virtual machine configuration can include a virtual transmission control unit (TCU). One real TCU can be used to simulate multiple virtual TCUs.

A real TCU cannot be shared by concurrently operating virtual machines. However, by defining <u>virtual</u> transmission control units, a real TCU can be concurrently shared by several virtual machines.

VIRTUAL CHANNEL-TO-CHANNEL ADAPTER

The user can define a virtual channel-to-channel adapter (CTCA) with or without a real equivalent. If a real channel-to-channel adapter is dedicated to a virtual machine, that virtual machine can communicate with a real computing system other than its own real system. However, 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) which is described briefly in this section under the topic "Other System Features."

VM/370 Directory

The VM/370 directory is a file containing entries for all potential virtual machines. These entries contain control statements that define each virtual machine and its' configuration. It contains the user identification and password, as well as the CP command privilege class assigned to the user.

Other optional information can be included such as performance and operating parameters.

```
USER VIRT1 VIRT1 512K 1M G
ACCOUNT S5 SYSPRG
CONSOLE 009 3215
SPOOL 00C 2540 READER A
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
MDISK 191 3330 001 001 CPR6L1 R
LINK MAINT 194 194 RR
```

Figure 4. Sample Directory Entry

Figure 4 describes a virtual machine with a user identification of VIRT1, a password of VIRT1, a normal virtual storage size of 512K, and a maximum virtual storage size of 1M. The account number is S5 and the separator identification is SYSPRG. The virtual I/O device configuration consists of the system console at 009 and spooled unit record devices at virtual devices addresses 00C, 00D, and 00E. Spool output class A is assigned to all devices.

The LINK statement indicates this virtual machine has a read-only access to a minidisk defined to another virtual machine. One permanent 3330-series minidisk is also defined.

When a user gains access to the VM/370 system, a virtual machine is created based on the information in the user's directory entry.

LOGON PROCEDURE

To logon to VM/370, a user enters (via a keyboard terminal device that is connected to the VM/370 system) a valid userid and a password. This and other information in the user's directory enables VM/370 to create a virtual machine that is tailored to the individual user.

Figure 5 illustrates a logon procedure for user "USER1". The user enters the userid. The system responds and then waits for the password to be entered. (For security purposes this password is masked.) CP verifies the userid and password. The user is then notified by a CP response that he is logged on. After successfully logging on the system, the user may load any supported operating system and proceed.

```
vm/370 online
logon USER1
ENTER PASSWORD:

LOGON AT 11:03:18 ON WEDNESDAY 11/29/78

ipl cms
```

Figure 5. Logging onto VM/370 and Loading CMS

LOGON/AUTOLOG AND LINK JOURNALING

The security journaling option, specified during system generation, does not allow passwords for the LOGON/AUTOLOG and LINK commands to be entered on the same line as the command.

If the security journaling option is specified, and the logon attempt is unsuccessful, these attempts are recorded and the system administrator is notified via messages.

Virtual Machine Operating Systems

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

SUPPORTED OPERATING SYSTEMS

Figure 6 lists the operating systems that can execute in virtual machines.

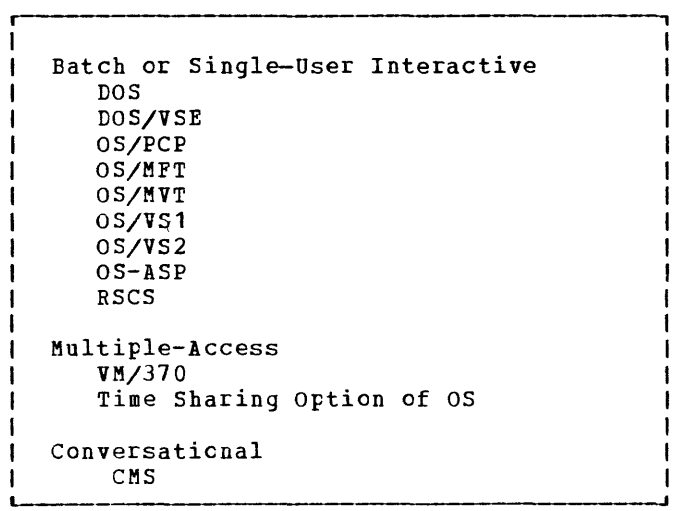


Figure 6. Virtual Machine Operating Systems

With the exception of OS/PCP, these are all multiprogramming systems. However, when operating in a virtual machine under VM/370, the user has the choice of running multiple partitions in one virtual machine (similar to stand-alone operation) or single partitions in multiple virtual machines. When running multiple partitions in one virtual machine, multiprogramming and unit record spooling is done by both the operating system and VM/370. When running single partitions in multiple virtual machines, the need for multiple virtual storages places a burden on auxiliary storage. However, this can be alleviated by using shared systems. Shared systems are discussed further on in this section.

Single-user systems

Systems that can be executed interactively by a single user include the conversational monitor system (CMS) 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.

Multiple-access systems

Multiple-access systems, such as VM/370, execute in one virtual machine and directly service many interactive terminals. To connect a terminal with the virtual machine, the user of a multiple-access system issues the DIAL command instead of the LOGON command.

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: When using the DIAL command, the user may not log onto CP. Therefore, the CP commands cannot be used unless the system logged onto is a virtual VM/370.

OTHER PROGRAMS AND SYSTEMS

For information about other programs and systems that have been used under VM/370, request information on Installed User Programs (IUPs) and Field Developed Programs (FDPs) from your local IBM branch office.

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.

SYSTEM PROGRAMMING

An interactive virtual machine environment provides these advantages for system programming:

- Reducing the amount of hands-on testing time on the real machine -- thereby reducing the requirements for off-shift testing
- Testing new or modified SVC routines in a virtual machine
- Applying and testing PTFs on a virtual machine

- Generating and testing in a virtual machine either new independent component releases (ICRs) or new releases of an operating system
- Debugging from a hands-on console (a terminal device) as though on a dedicated real machine

APPLICATION PROGRAMMING

An interactive virtual machine environment provides these advantages for application programming:

- Using the CMS editor to create source programs and data files
- Using the CMS UPDATE command and the CMS editor to maintain source programs and data files
- Debugging from a terminal while under operating system control
- Providing faster turnaround time, more test periods per day, and a shorter development cycle
- Designing application programs without real storage limitations
- Defining minidisks and other virtual devices to design and test a slightly different or larger machine configuration before installing the hardware
- Using SCRIPT/370, an Installed User Program (IUP) for text preparation, to create and update program specifications

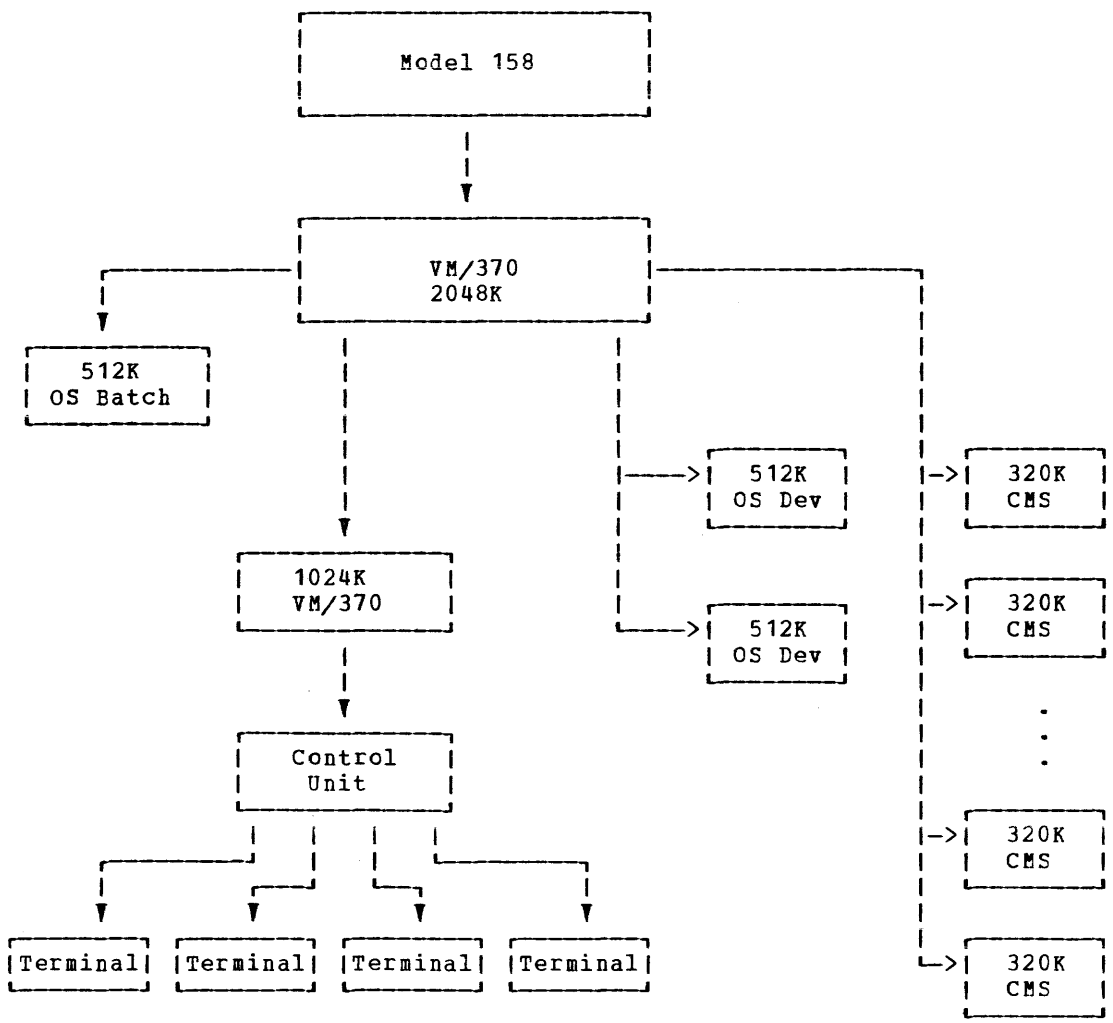


Figure 7. Virtual Machines Running Concurrent Production, Development, and Testing

OPERATIONS

An interactive virtual machine environment relieves problems of scheduling, support, and backup and expedites production. Some of these operational advantages are:

- Training operators in a virtual machine that is isolated from production virtual machines
- Providing a shorter training period
- Defining a virtual machine and its devices as backup to another real machine
- Running different types of work concurrently on a single real machine
- Reducing the manual handling and scheduling of user test periods by the operations staff -- users run their own virtual machines
- Executing many types of batch applications with no change to the program either in an individual virtual machine or in a virtual machine dedicated to executing programs in batch mode.

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.

Backup System

An installation using VM/370 has more flexibility in using another System/370 computing system for backup. Neither the same System/370 model nor the same amount of real storage have to be part of the backup system.

The backup system must include, but is not limited to, the same type and number of real devices as these virtual machines require. Also, 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 installation need not be concerned with device addresses.

Performance Options

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

VM/370 offers the following performance options to enhance the operating efficiency of one or more virtual machines. The performance options are:

- Virtual machine assist
- VM/370 Extended Control-Program support
- Locked pages
- Reserved page frames
- Virtual=real
- Favored execution
- Priority
- Affinity in an attached processor environment

For detailed information concerning these performance options, refer to VM/370 Operator's Guide and VM/370 Planning and System Generation Guide.

VIRTUAL MACHINE ASSIST

When an operating system is run in a virtual machine under VM/370 it is dispatched by CP in the problem state. All privileged instructions issued by the operating system cause program interrupts and CP must simulate the desired effect of the operation. The virtual machine assist feature performs many of the privileged instruction simulation functions without the virtual machine leaving the problem state. The result is a considerable reduction in overhead. Virtual machine assist may be considered a prerequisite to running production jobs under an operating system in a virtual machine.

The system operator enables or disables the virtual machine assist via a command. However, if the function has been enabled for the system, the user has the choice of setting it off for his machine, and later setting it on again.

VM/370 EXTENDED CONTOL-PROGRAM SUPPORT

VM/370 Extended Control-Program support is a hardware assist function that is available on certain VM/370 supported processors. This hardware assist function, when used with the virtual machine assist function, further reduces VM/370 real time to support virtual machines, thus adding improved performance to the VM/370 system.

LOCKED PAGES

The system operator can, by use of a command, lock certain virtual storage pages into real storage. This is done to eliminate paging activity on these pages, thus reducing virtual storage.

RESERVED PAGE FRAMES

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. The pages are not locked into these page frames. Pages can be paged out, but only for other active pages of the same virtual machine.

VIRTUAL=REAL

The virtual=real option can be assigned during system generation. With this option, a virtual=real area is allocated directly from storage when VM/370 is initially loaded, and that area remains allocated unless it is released by the system operator.

The virtual=real option improves performance in the selected virtual machine because CP no longer has to perform paging operations.

PRIORITY OPTION

The priority option affects the execution of a particular virtual machine as compared with other virtual machines that have the same characteristics. A priority value assigned to a virtual machine is used, in combination with other factors, to influence the dispatching algorithm. The lower the value, the greater the percentage of processor time given the virtual machine.

FAVORED EXECUTION

The favored execution option provides a particular virtual machine an assured percentage of real processor time.

AFFINITY

Under the normal attached processor, either processor can handle the virtual machine execution. However, with the affinity capability, only the specified processor will allow virtual machine execution.

PERFORMANCE MEASUREMENT AND ANALYSIS

The VM/370 control program has two commands that cause CP to collect and display performance related data.

The MONITOR command gathers data relating to most aspects of system performance and writes the data on tape or spool file. When the data is reduced, 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.

Error Recording and Analysis

Operating systems that commonly run in virtual machines write error records to the error recording data sets. However, in a virtual machine CP intercepts the error, and VM/370 records the error in its own error recording cylinders. Therefore, error records from all operating systems reside in this one centralized error recording area.

Other System Features

Besides error recovery and CP command features, VM/370 contains several other features that expand the capabilities of operating systems running in virtual machines. They are:

- Virtual machine accounting
- Saved systems
- Shared systems

- Discontiguous saved segments
- Shared segment protection
- Virtual machine communication facility

VIRTUAL MACHINE ACCOUNTING

VM/370 keeps track of a virtual machine's usage of system facilities and punches accounting cards whenever the use of some chargeable resource is terminated. For operating systems that do not have job accounting, this would be one way of having VM/370 gather job accounting data.

When accounting data records are generated, CP stores them in real storage. These records remain there until they are punched on the real punch.

SAVED SYSTEMS

When initially loading an operating system into a virtual machine by device address, VM/370 reads the resident nucleus into real storage and writes it back out to the system paging device. Simultaneously it updates the virtual machine's paging tables.

To save the results of an IPL at a predetermined point (by the type of operating system), VM/370 allows an installation to assign this saved system a name.

Note: Saved systems cannot be loaded into the virtual=real area of VM/370. An attempt to do so results in an error message.

SHARED SYSTEMS

A saved system can also share reentrant portions of its virtual storage among many concurrently operating virtual machines.

In a uniprocessor environment, regardless of the number of virtual machines using the shared segments, only one copy of the pages in the shared segment need ever occupy real storage and external page storage. In an attached processor environment, there will be one copy of the page(s) in the shared segment per processor. This reduces the total real storage and auxiliary storage required.

The greater the number of virtual machines that are using a shared system, the greater the storage savings, and the greater the probability that the shared page(s) will be frequently referenced. Frequently referenced pages tend to remain in real storage, thereby reducing paging activity. Less paging activity increases the efficiency of the processor.

DISCONTIGUOUS SAVED SEGMENTS

Discontiguous saved segments (DCSS) are similar in many ways to saved systems. DCSS must have been named, loaded, and saved as when saving a system; they must also be loaded at an address higher than the highest address of the virtual machine using it.

DCSS can be logically attached by a virtual machine whenever needed and detached when not needed. The CP instructions required to perform the linkage must be part of the virtual machine's operating system.

Note: Systems using DCSS cannot be loaded into the virtual=real area of VM/370. An attempt to do so results in an error message.

SHARED SEGMENT PROTECTION

The mode of shared segment protection may be specified at system generation time.

With shared segment protection, VM/370 protects the multiple users of shared segments from any intentional or inadvertent alteration of the shared segment's code or data. Before CP selects a new user to be dispatched, it checks whether or not the current virtual machine has altered any pages in shared segments. If so, CP places the virtual machine in console function mode, returns the altered page to free storage, and prepares a fresh copy of the shared page for the remaining shared system users.

Without shared segment protection specified for a named system, changes made by a user to pages in shared segments of the system remain undetected by the VM/370 control program. All users of the unprotected shared segment will see the change.

Installations choosing to run with unprotected shared segments should carefully evaluate these segments to ensure that they contain read-only information. The unprotected option for shared segments will eliminate the overhead involved in scanning for changed pages in both uniprocessor and attached processor systems, in switching page table pointers, and in maintaining duplicate copies of shared systems in an attached processor environment.

VIRTUAL MACHINE COMMUNICATION FACILITY

The virtual machine communication facility (VMCF) allows one virtual machine to communicate and exchange data with other virtual machines operating under the same VM/370 system. A more detailed description of VMCF can be found in the "Control Program" section of this publication.

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 specific amounts of virtual storage and 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 processor that is actually doing the work.

VM/370 uses a technique called time slicing so that one real processor appears to be multiple virtual processors. Each virtual machine periodically gains access to the real processor 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 processor 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 processor 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 processor 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

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 megabytes. 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.

OPERATING SYSTEM

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 processor. 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.

SEGMENTS

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.

The protected or unprotected status of a system containing shared segments is specified at system generation. For a protected shared segment, if a user attempts to alter a page in a shared segment the user receives a message informing him of an integrity violation and his virtual machine is placed in console function mode. For both protected and unprotected shared segments: Whenever a user issues a command that alters a shared segment, the user is unshared from the shared segment and given a private copy of the segment including the altered page.

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.

PAGING

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.

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. Paging operations are initiated and performed by CP and require no action by the virtual machine.

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 during CP program execution.

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 one with available space. If the page is not changed, it remains allocated in its original direct access locations and is paged into real storage from there the next time the virtual machine refers to that page.

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.

When an I/O device is a Mass Storage System (MSS) 3330V, but the virtual device is defined as a 3330-1, CP places a prefix on the actual channel program. This prefix allows CP to determine when an MSS cylinder fault occurs and queue the I/O operation until the cylinder fault is corrected. This operation is transparent to the virtual machine. However, if the virtual device is defined as a 3330V, no prefix is added to the channel program and cylinder faults must be processed by the virtual machine.

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.

Interrupts indicating that an MSS cylinder fault has occurred, and that a cylinder fault has been resolved, are passed to the virtual machine only if the virtual device is defined as a 3330V.

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

Virtual disks, often called minidisks, 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.

CHANNEL

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.

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 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.

DIAGNOSE INTERFACE

A virtual input/output operation by CP can be simplified if the virtual machine uses the DIAGNOSE interface. CMS was designed specifically for the virtual machine environment, and 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

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

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.

ADDING AND TRANSFERRING FILES

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. See Figure 8 for a concise description of each privilege class.

GENERAL USERS

A general user can issue commands that control functions associated with the execution of his virtual machine.

A complete list of the general user commands and their descriptions, are located in the VM/370 CP Command Reference for General Users.

Class	User	Function
A[1]	Primary System Operator Note: The class A user who is logged on at CP initialization is designated as the primary system operator.	To be responsible for the availability of the VM/370 system, its communication lines, and resources. In addition, the class A user controls system accounting, broadcast messages, virtual machine performance options and all other commands that affect the overall performance of VM/370.
B[1]	System Resource Operator	To control all the VM/370 real resources, except those controlled by the class A or class D user.
C[1]	System Programmer	To update certain functions not controlled by other privileged class users.
D[1]	Spooling Operator	To control all spooling data and specific functions of the unit record devices.
E[1]	System Analyst	To examine and save specific data in the VM/370 storage area.
F[1]	Service Representative	To obtain and examine, in detail, certain data concerning the I/O devices connected to the VM/370 system.
G[2]	General User	Control functions associated with the execution of the users virtual machine.
Any[2]	Any User	Gain and relinquish access to the VM/370 system.
H	Reserved	Reserved for IBM use only.

[1]Described in the VM/370 Operator's Guide
[2]Described in the VM/370 CP Command Reference for General Users

Figure 8. CP Privilege Class Descriptions

OTHER USERS

Users, other than the general user, can perform additional functions by issuing commands to dynamically provide VM/370 performance options, to terminate a particular virtual machine and many other similar functions.

VM/370 Attached Processor Support

On systems that are either an attached processor complex or an asymmetric multiprocessor complex, VM/370 has a mode of operation that takes advantage of the additional computing power that such systems provide. This mode of operation is called attached processor mode. In this mode, two instruction processors simultaneously execute multiple tasks from a common queue while sharing the same main storage.

With attached processor mode, the main processor controls and performs all I/O activity. In the event of a severe nonrecoverable malfunction in the attached processor hardware, VM/370 may continue operations with the main processor executing in uniprocessor (or one processor) mode. (Note, the main processor is the processor equipped with channel and IO devices.) If the system complex supports Channel-set Switching, and a severe nonrecoverable hardware error is associated with the main processor, the

system may still be able to perform in uniprocessor mode. It does this by dynamically switching the channel set (all active channels of the main processor) to the other processor, thereby, utilizing the attached processor as the main processor.

This flexibility of system operation, along with CP's VARY ON/OFF PROCESSOR command, facilitates system reliability, availability and serviceability (RAS). RAS is enhanced because main components of a system complex can be placed offline for system repair or upgrade at a time that minimally impacts system operations. Later, when the full resources of the system are needed and available, the processor can be varied online and attached processor mode reestablished.

Virtual Machine Communication Facility

The virtual machine communication facility (VMCF) allows one virtual machine to communicate and exchange data with any other virtual machine operating under the same VM/370 system.

This is accomplished via a DIAGNOSE instruction utilizing a special parameter list. VMCF locks one data page for the sender and one for the receiver during the transfer. As a result, data transfer is faster because VMCF moves information directly from the sender's virtual storage to the receiver's virtual storage. The transfer does not involve devices because VMCF uses userids to logically address data from one virtual machine to other virtual machines.

A virtual machine can have one to 50 active communications going on with other virtual machines. The amount of data that can be transferred in a single operation is limited only by the virtual storage size of the virtual machines involved in the transaction.

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.

VM/VS Handshaking

VM/VS handshaking is a communication path between the control program (CP) component of VM/370 and the following operating systems running as virtual machines under VM/370:

- DOS/VS Release 34 Advanced Functions -- DOS/VS program product (program number 5746-XE2)
- VS1 Release 4 and subsequent releases

To improve their operation with VM/370, the operating system and CP make each other aware of mutual capabilities or requirements. Systems generated to use VM/VS handshaking may have the dual ability to run both in a real machine and in a virtual machine. Virtual machine systems that have VM/VS handshaking can more realistically simulate the operation of their real machine.

VM/VS handshaking consists of:

- Closing CP spool files when job output is complete. This allows VM/370 to immediately process these output files without operator intervention.
- Processing pseudo page faults. When the pseudo page fault handling portion of handshaking is active, one task can be dispatched while another is waiting for a page to be brought into real storage.
- Providing a non-paging mode to eliminate duplicate paging.
- Providing a way to avoid a PCI (program-controlled interruption) in a BTAM autopoll CCW loop.
- Providing miscellaneous enhancements when running under VM/370.

Conversational Monitor System

The conversational monitor system (CMS) is a major 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.)

The user can create, update, and manipulate files as well as complete, test, and execute problem programs. These interactive capabilities are extended to DOS/VS users via the CMS/DOS environment or CMS. For OS/VS users, a combination of CMS commands and CMS simulation of OS macros provides similar interactive capabilities.

CMS Configuration

A virtual machine that is to use CMS is configured much the same way 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 megabytes, 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.

Figure 9 shows a virtual machine configured to execute CMS. A minimum CMS configuration would not include the virtual tapes.

CMS File System

CMS supports the same DASD devices that are supported by the CP portion of VM/370. CMS formats the tracks of a CMS disk into fixed-length blocks. The CMS file system then manages these blocks in such a way that the user appears to have logical fixed- or variable-length records, and sequential or direct access to files.

In addition to reading and writing CMS files, CMS can read sequential or VSAM DOS files and sequential, partitioned, or VSAM OS 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.

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 invoked by CMS commands also manipulate files. CMS files can be written onto and restored from unlabeled

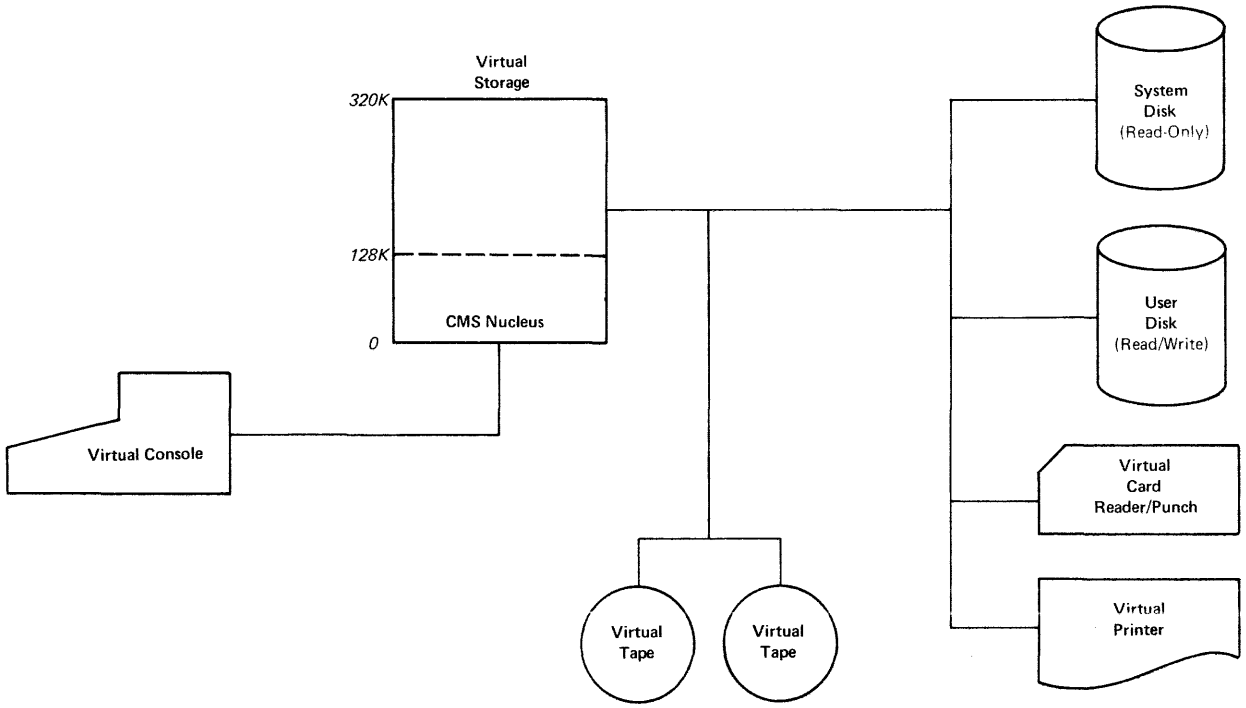


Figure 9. Sample CMS Configuration

tapes via CMS commands. Tape labels are not supported by CMS.

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.

USER DISKS

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.

Each virtual disk may be defined as read-only or read/write, and may be shared among users.

FILE IDENTIFICATION

CMS user files are identified by a fileid consisting of three designators:

- filename (fn) -- is an 8-character name assigned the file by the user.
- filetype (ft) -- is the identification given to specify particular file characteristics.
- filemode (fm) -- describes the disk that contains the file and the access mode assigned to the file.

```
SAMPLE  ASSEMBLE  A1
```

Figure 10. CMS File Identification

Figure 10 is a sample CMS file with an identification of SAMPLE, ASSEMBLE indicates that the file consists of assembler language source statements, A indicates that the file is located on the A-disk directory, and 1 indicates that the file has a read/write access mode.

FILE STRUCTURE

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 has a master file directory, or the subset called user file directory, containing format and size information for each file on the virtual disk. This directory includes a pointer to the file's chain link records.

Figure 11 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.

For 800-byte record disks, a single user file is limited to a maximum of 65,533 records and must reside on one virtual disk. The file management system limits the number of files on any one virtual disk to 3400 (3500 files on a 2314). All CMS disk files are written as 800-byte records, chained together by a specific file entry that is stored in a table called the master file directory; a separate master file directory is kept for, and on each virtual disk.

Initialization and Dump Restore

The OS IBCDASDI service program initializes all types of real and virtual minidisks supported by VM/370.

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 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 on the terminal the program name, followed by any required operands.

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 rekeying 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.

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

PROGRAM COMPILATION AND EXECUTION COMMANDS

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.

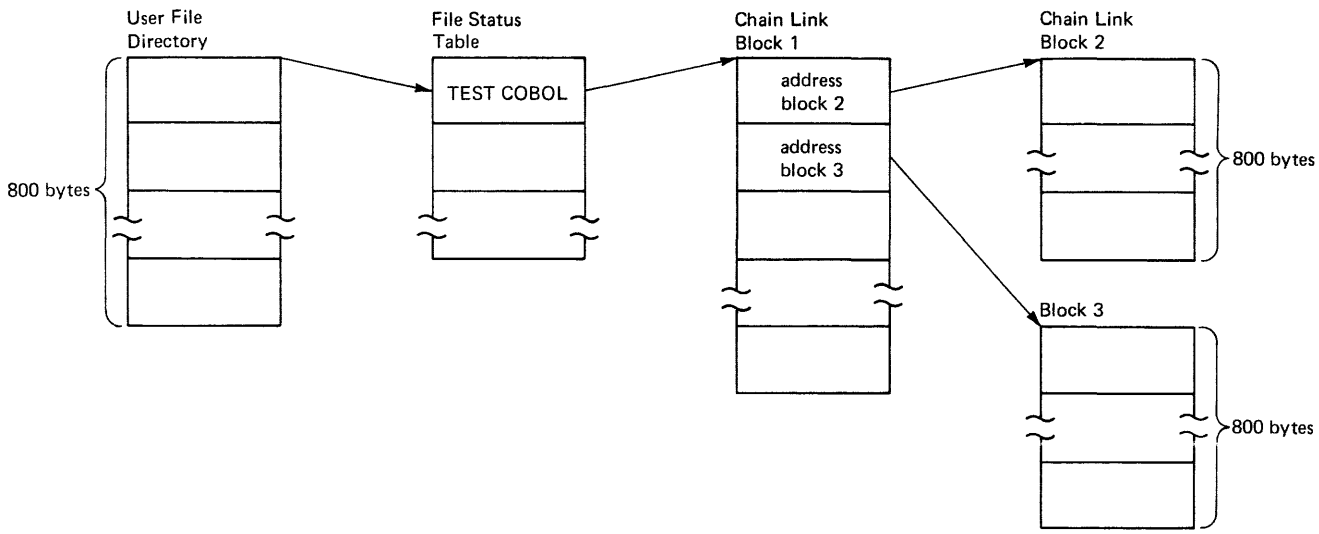


Figure 11. CMS File Structure

The compilers executable under CMS are invoked by name and provided with a source file whose filetype designator indicates the compiler. On each of the 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. 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.

The DOS linkage editor is simulated in CMS. Files to be link-edited can be read from DOS libraries. CMS link-edits files and places the output in a CMS file called DOSLIB and accepts the DOS linkage editor control statements as input.

LANGUAGE PROCESSORS

A VM/370 assembler is distributed as part of the VM/370 system and is required for installation and support. 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, VSBASIC, DOS and OS PL/I, and OS FORTRAN are useful languages for problem-solving applications. DOS/VS and OS/VS COBOL, DOS and OS PL/I and assembler languages are useful for commercial program development applications.

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

INTEGRATED EMULATORS

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

No modifications are required to be made to emulators, to DOS or OS, or to VM/370 to allow emulator-dependent programs to execute in virtual machines.

In an attached processor (AP) system, a virtual machine can use the SET AFFINITY command to make use of an emulator installed on only one of the processors. The directory option for affinity may be used instead, with similar results.

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.

SCRIPT/370

SCRIPT/370 is a text processor available as an IUP (Installed User Program) for use under CMS. SCRIPT/370 includes manuscript facilities that create formatted output from one or more CMS files containing text and/or text-manipulating control words.

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.

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.

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.

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 virtual machine or the real card reader to a virtual machine dedicated to running batch jobs under the batch facility. The batch facility virtual machine then executes these jobs, freeing the user's virtual machine for other uses. The accounting routines charge the time used in the batch machine to the originating user.

Remote Spooling Communications Subsystem

The VM/370 remote spooling communications subsystem (RSCS) is the VM/370 component that runs under CP control. RSCS provides telecommunication support to transfer files between remote stations and virtual machines using binary synchronous (BSC) switched or nonswitched lines.

RSCS cannot be operated without CP. RSCS, together with the spool file system of CP, provides a telecommunication environment for the transfer of files from:

- Virtual machines to RSCS remote stations
- RSCS remote stations to other RSCS remote stations
- Virtual machines to remote job entry batch system
- RSCS remote stations to job entry batch systems
- RSCS remote stations to a CMS batch virtual machine

Some examples of remote job entry batch systems are: HASP, ASP, JES, RES, and CRJE. These job entry batch systems, run under the appropriate system control program such as: DOS, DOS/VSE, OS, or OS/VS.

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.

For details concerning RSCS refer to the VM/370 Remote Spooling Communications Subsystem (RSCS) User's Guide.

Figure 12 shows the relationship among 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 processor, transmission control units, and BSC telecommunications lines and remote stations.

THE REAL PROCESSOR

A real processor 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.

REMOTE STATIONS

RSCS remote stations are input/output (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 though it were nonprogrammable.

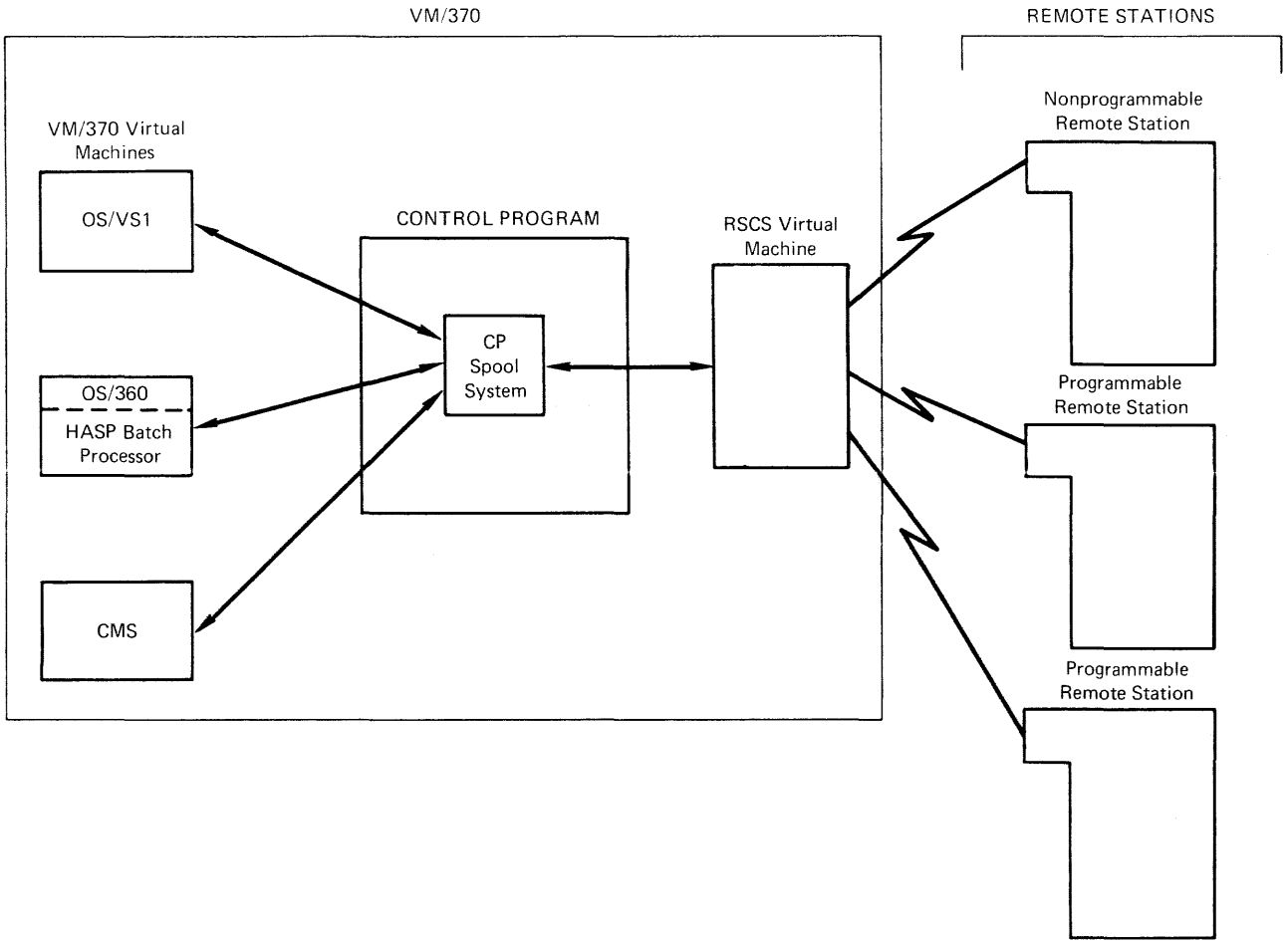


Figure 12. A VM/370 RSCS Teleprocessing Network

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.

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.

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.

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 stations.

REMOTE SPOOLING COMMUNICATIONS SUBSYSTEM NETWORKING

The Remote Spooling Communications Subsystem Networking program product (Program Number 5748-XP1) provides enhanced networking facilities to a VM/370 system.

RSCS Networking is a complete virtual machine subsystem that operates independently of other operating systems running under CP.

For further information concerning RSCS Networking, refer to the RSCS Networking General Information Manual, GH24-5004 and the RSCS Networking Program Reference and Operations Manual, SH24-5005.

Interactive Problem Control System

The interactive problem control system (IPCS) is a component of VM/370 that standardizes the VM/370 problem reporting process. It enhances the serviceability of VM/370 by identifying recurring programming problems located on the VM/370 user base. Additional environmental conditions can also be included in problem reports; user detected problems can be entered, updated, and printed out; and the problem data base can be interrogated for individual problems. The function of generating problem reports is accomplished by the VMFDUMP command and by providing an additional command called PROB. Both generate a disk-resident report. This report, along with the early warning system (EWS) microfiche, gives speedier identification of previously reported problems.

The IPCS component:

- Facilitates online problem management and interactive problem diagnosis.
- Provides an analysis function for CP dumps; the resultant extracted data resides on disk as a unique problem report.
- Has a prompting program for user-detected failures; the resultant data resides on disk as a unique problem report.
- Searches for duplicate problems and informs the user when a problem with similar symptoms was previously experienced at the installation.
- Has a problem-tracking facility that updates and displays problem status.
- Has an interactive debugging facility that aids in the analysis of disk-resident CP abend dumps.

Conditions of Operation

The amount of available IPCS virtual machine A-disk space can affect the operation of IPCS.

When an adequate amount of storage is available, all problem reports, dumps, supplementary data files, and the symptom summary file are stored on disk. When storage is at a premium, only the problem reports and symptom summary file are kept on disk.

More information concerning the operation of IPCS can be found in the VM/370 Interactive Problem Control System (IPCS) User's Guide, GC20-1823.

PROBLEM REPORT GENERATION

IPCS can be used to report two types of problem situations. They are:

- CP system detected failures
- All user-detected failures

IPCS COMMANDS

IPCS commands are invoked from a CMS virtual machine. They are:

DUMPSCAN
: provides the system analyst edit facilities to obtain, via the terminal console, pertinent data from a system dump for quick analysis.

PRE
: provides a means of updating the status of reported problems.

PROB
: provides the user with a means of entering problems into the IPCS reporting system that are not associated with CP abend dump conditions.

STAT
: provides a means of interrogating the problem report files.

VMFDUMP
: provides a formatted dump and a system generated problem report for the IPCS system.

PRELIMINARY CONSIDERATIONS

Problem determination and resolution often require analysis of other data in addition to CP abend dumps. This data is called supplementary data in the IPCS system. Procedures must be established for

collecting data prior to the problem situation.

Some methods used to collect data are:

- Spooling of the system operator's console
- Virtual machine dumps
- Trace output

For more detailed information regarding IPCS, refer to VM/370 Interactive Problem Control System (IPCS) User's Guide.

INSTALLING IPCS

The installation of CMS includes the IPCS component of VM/370: the VM/370 Planning and System Generation, GC20-1801 details the system generation procedures.

Appendix A: VM/370 Publication-to-Audience Relationship

The following illustrates the relationship of the VM/370 publications to their appropriate audience and also includes a brief description of the text content.

Audience	Publications	Content
All users	Introduction, GC20-1800	Overview of VM/370 and virtual machines
	CP Command Reference for General Users, GC20-1820	G and "Any" commands — needed for virtual machine control
	System Messages, GC20-1808	Abend codes, wait states, and messages
	Terminal User's Guide, GC20-1810	Logon/Logoff, virtual, console terminal use, attention handling
	Quick Guide for Users, GC20-1926	Reference for command and service program control statement formats
System Analyst, Programming System Representatives, Installation Planners and Support Personnel	Release 6 Guide GC20-1834	VM/370 planning information for Release 6
	Planning and System Generation Guide, GC20-1801	VM/370 installation requirements and procedures
	Operator's Guide, GC20-1806	VM/370 system control commands, resources, control, service programs, spooling
	System Programmer's Guide, GC20-1807	Resources for problem solving, tracking, tracing, monitoring
	Operating Systems in a Virtual Machine, GC20-1821	Relationship of VM/370 with other operating systems, VM/370 concepts
	OLTSEP and Error Recording Guide, GC20-1809	OLTS from a virtual environment. RMS, invoking CPEREP
	CMS Command and Macro Reference, GC20-1818	CMS facilities for generating, compiling editing, testing and debugging programs
	CMS User's Guide, GC20-1819	
	OS/VS, DOS/VSE, VM/370 Environmental Recording Editing and Printing, GC28-0772	Provides details on OS/VS, DOS/VSE, and VM/370 EREP operands invoked by CPEREP

Audience	Publications	Content
System Analyst, Programming System Representatives, Installation planners and Support Personnel (cont.)	Interactive Problem Control System (IPCS) User's Guide, GC20-1823	Catalog and update facility for recorded system and virtual machine problems
	Service Routines Program Logic, SY20-0882	VM/370 logic on CP, CMS and RSCS; the service programs; OS/VS EREP and the data areas and control blocks that relate to those components
	Data Areas and Control Block Logic, SY20-0884	
	System Logic and Problem Determination, Vol 1 - CP, SY20-0886 Vol 2 - CMS, SY20-0887 Vol 3 - RSCS, SY20-0888	
	OS/VS, DOS/VSE, VM/370 Environmental Recording Editing and Printing (EREP) Program Logic, SY28-0773	
Field Engineer (CE)	OLTSEP and Error Recording Guide, GC20-1809	OLTS from the virtual machine, RMS, and Invoking CPEREP
	OS/VS, DOS/VSE, VM/370 Environmental Recording Editing and Printing GC28-0772	Provides details on OS/VS DOS/VSE and VM/370 EREP operands invoked by CPEREP
System Operator	Operator's Guide, GC20-1806	Resource and system control command and service programs; IPL VM/370
	Remote Spooling Communications Subsystem (RSCS) User's Guide, GC20-1816	Operational control of remote spooling
	CP Command Reference for General Users, GC20-1820	G and any commands — Needed for CP virtual machine CP control
	Operating Systems in a Virtual Machine, GC20-1821	Relationship of VM/370 with other operating systems Expands the information on G command usage
	Interactive Problem Control System (IPCS) User's Guide, GC20-1823	Tracking facility for system and virtual machine problems
	OLTSEP and Error Recording Guide, GC20-1809	How to invoke CPEREP to capture/clear SYS1.LOGREC and VM/370 error cylinders

Audience	Publications	Content
Application Programmer, Interactive users — Anyone who needs to create or modify programs or data	Release 6 Guide GC20-1834	VM/370 Planning information for Release 6
	CMS Command and Macro Reference, GC20-1818	CMS command formats for program creation, editing testing and debugging
	CMS User's Guide, GC20-1819	Detailed use of CMS commands

Appendix B: 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 that General User)[1]	GX20-1995
VM/370 CMS Command and Macro Reference	GC20-1818
VM/370 CMS User's Guide	GC20-1819

Coreguisite 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 User's Guide Program No. 5796-PH6	SH20-1857
SCRIPT/370 Systems Guide	LY20-2251
SCRIPT/370 Quick Guide for Users Reference Summary	GX20-1997

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. GBOF-3576.

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

COBOL User

Note: "IBM OS Full American National Standard COBOL" has been abbreviated to "OS ANS COBOL" in the 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 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/VSE VSAM and CMS VSAM Users

DOS/VSE Macro User's Guide	GC24-5139
DOS/VSE Macro Reference	GC24-5140
Using VSE/VSAM Command and Macros	GC33-5382
DOS/VSE Data Management Concepts	GC24-5138

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
DOS PL/I Transient Library Program Product Specifications	GC33-0018

VS APL User

VS APL: Terminal User's Guide for CMS	SH20-9067
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 Reference	GC26-3847

Index

READER'S COMMENTS

Title: IBM Virtual Machine Facility/370: Introduction

Order No. GC20-1800-9

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

Reader's Comment Form

Cut or Fold Along Line

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE:

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

Attn: VM/370 Publications

Fold and tape | Please Do Not Staple | Fold and tape

IBM Virtual Machine Facility/370: Introduction

Printed in U.S.A.

GC20-1800-9

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

GC20-1800-9

IBM Virtual Machine Facility/370: Introduction Printed in U.S.A. GC20-1800-9

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

Technical Newsletter

This Newsletter No. GN25-0489
Date August 1, 1979

Base Publication No. GC20-1800-9
File No. S370-20 (VM/370 Release 6 PLC 4)
Prerequisite Newsletters/ Supplements None

IBM Virtual Machine Facility/370: Introduction

This Technical Newsletter contains replacement pages for VM/370 Introduction to support Release 6 PLC 4 of IBM Virtual Machine Facility/370.

Before inserting any of the attached pages into the VM/370 Introduction, read carefully the instructions on this cover. They indicate when and how you should insert pages.

Pages to be Removed	Attached Pages to be Inserted*
Title, Edition Notice	Title, Edition Notice
Table of Contents vii-viii	Table of Contents vii-vii
Summary of Amendments ix-xiv	Summary of Amendments ix-xiv
17-18	17-18.2

*If you are inserting pages from different Newsletters/Supplements and identical page numbers are involved, always use the pages with the latest date (shown in the slug at the top of the page). The page with the latest date contains the most complete information.

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

Summary of Amendments

This Technical Newsletter incorporates changes reflecting Channel-set Switching support.

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

IBM Corporation, Publications Development, Department D58, Building 706-2,
PO Box 390, Poughkeepsie, New York 12602

Printed in U.S.A.