GC19-6200-1
File No. S370/4300-20

Program Product

IBM Virtual Machine/ System Product: Introduction

IBM

GC19-6200-1
File No. S370/4300-20

Program Product

IBM Virtual Machine/ System Product: Introduction

Program Number 5664-167

Release 2

IBM

Second Edition (April 1982)

This edition, GC19-6200-1, is a revision of GC19-6200-0. It applies to the IBM Virtual Machine/System Product (5664-167) until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information contained herein; before using this publication in connection with the operation of IBM systems, consult the IBM System/370 and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

Summary of Amendments

For a list of changes, see page iii.

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

It is possible that this material may contain reference 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 IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Programming Publications, Department G60, P.O. Box 6, Endicott, New York, U.S.A. 13760. 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.

Summary of Amendments for
VM/SP Release 2

This revision includes changes for:

- IBM 3375 Direct Access Storage
- Trace Table Recording Facility

It also includes minor technical and editorial changes.

Preface

This publication introduces and describes the IBM Virtual Machine/System Product (VM/SP). The subjects described in this book are: Control Program (CP), Conversational Monitor System (CMS), remote spooling communications and interactive problem control.

This publication contains five sections and two appendixes:

- "Introduction" describes VM/SP, virtual machines, and their applications.
- "Control Program" describes how the VM/SP control program manages the resources of the real computing system. It also describes the automated operator support.
- "Conversational Monitor System" describes the facilities of CMS: problem solving and program development capabilities for interactive users.
- "Remote Spooling Communications" briefly describes the functions provided with a remote spooling network.
- "Interactive Problem Control Systems" briefly describes the functions provided by systems that generate and catalogue system malfunctions.
- "Appendix A: VM/SP Publication-to-Audience Relationship"
- "Appendix B: VM/SP-Related Publications for CMS Users and Other Applications"

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/System Product:

General Information Manual, GC20-1838

CMS Command and Macro Reference, SC19-6209

CMS User's Guide, SC19-6210

Commands (General User), SX20-4401 (reference card)

Commands (Other than General User), SX20-4402 (reference card)

CP Command Reference for General Users, SC19-6211

EXEC 2 Reference, SC24-5219

System Product Editor User's Guide, SC24-5220

System Product Editor Command and Macro Reference, SC24-5221

Library Guide and Master Index, GC19-6207

Operating Systems in a Virtual Machine, GC19-6212

Operator's Guide, SC19-6202

Planning and System Generation Guide, SC19-6201

Quick Guide for Users, SX20-4400 (pocket reference)

System Messages and Codes, SC19-6204

System Programmer's Guide, SC19-6203

Terminal User's Guide, GC19-6206

Data Areas and Control Block Logic, LY20-0891

System Logic and Problem Determination Guide Volume 1 Control Program (CP), LY20-0892

System Logic and Problem Determination Guide Volume 2 Conversational Monitor System (CMS), LY20-0893

Information concerning the Device Support Facilities program used with VM/SP to perform various inspection and initialization operations on direct access storage devices is found in:

Device Support Facilities User's Guide and Reference, GC35-0033

IBM Virtual Machine Facility/370:

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

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

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

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

IBM Virtual Machine/System Product OLTSEP and Error Recording Guide, SC19-6205

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/SP. The second publication provides detailed information on the operands that are common to OS/VS, DOS/VSE, and VM/370 as well as VM/SP.

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/System Product Service Routines Program Logic, LY20-0890

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.

The following communications provide information about the VTAM Communications Network Application VM/VCNA) Program Product.

IBM VM/VCNA General Information, GC27-0501

IBM VM/VCNA Installation, Operation, and Terminal Use, SC27-0502

MISCELLANEOUS

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

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

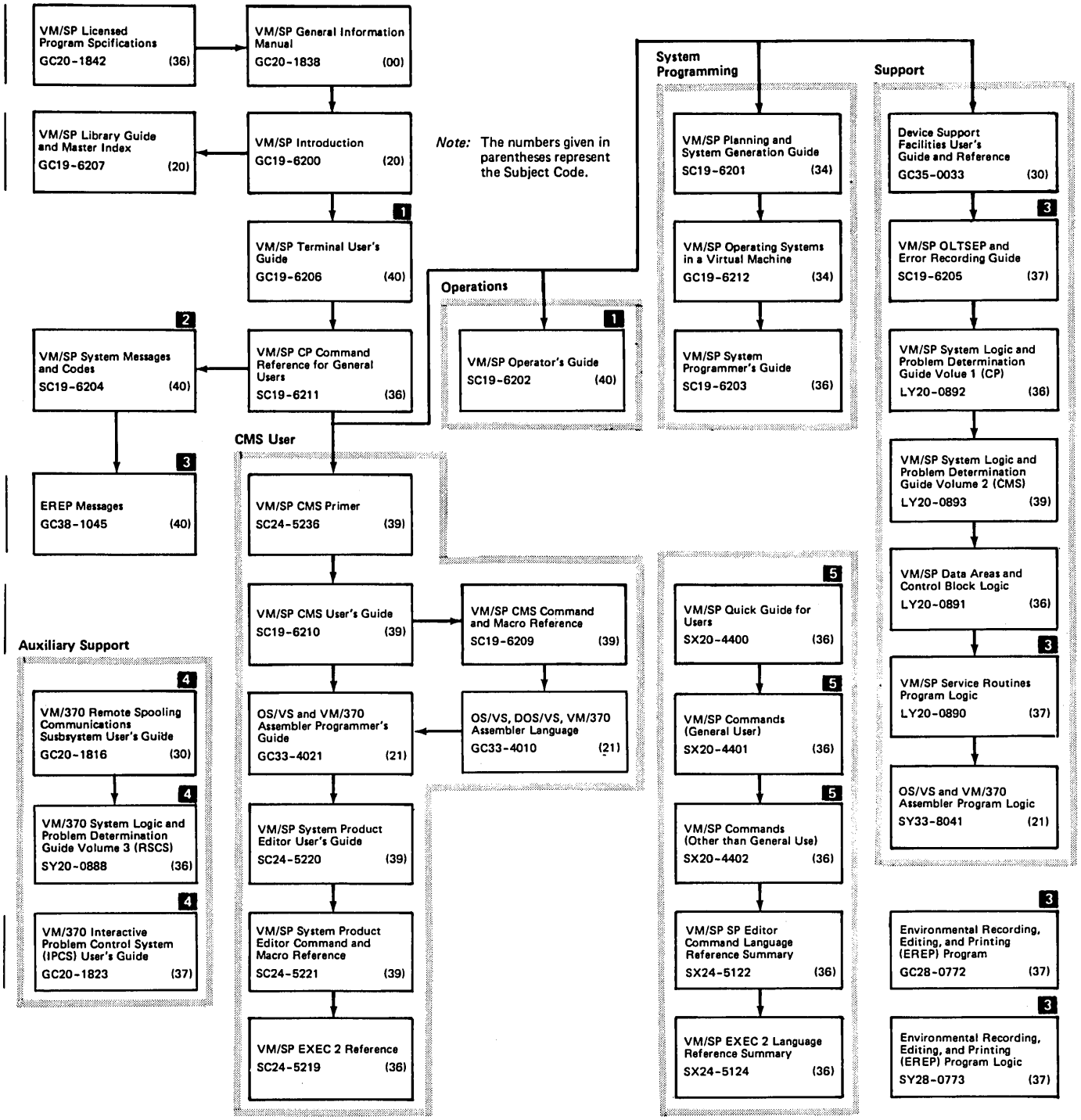
References in the text to titles of related VM/SP and VM/370 publications are given in abbreviated form.

Unless otherwise noted, the term VSE refers to the combination of the DOS/VSE system control program and the VSE/Advanced Functions program product.

In certain cases, the term, DOS, is still used as a generic term. For example, disk packs initialized for use with VSE or any predecessor DOS or DOS/VS system may be referred to as DOS disks.

The DOS-like simulation environment provided under the CMS component of VM/SP, continues to be referred to as CMS/DOS.

Publications that support VM/SP as used in conjunction with VM/370 Release 6



Legend:

1 For SNA terminal users, the prerequisite publication is: *VM/VCNA Installtion, Operations, and Terminal Use*, Order Number SC27-0502.

2 All users of virtual machine must use the *VM/SP System Messages and Codes* publication.

3 Contains information on VM/EREP support. EREP Release 3 is recommended for use with VM/SP Release 2.

4 VM/370 Release 6 components. However, the IPCS Extension Program Product (5748-SA1) and the RSCS Networking Program Product (5748-XP1) are recommended for use with VM/SP.

5 If you want all three of the Reference Summary publications, use SBOF 3820 when ordering.

Figure 1. Virtual Machine/System Product Library, and Related Publications

Contents

FIGURES

Virtual Machine/System Product (VM/SP) control program that manages the resources of an IBM System/370, or 4300, 3081, or 303x system complex 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 shared either 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/SP -Its Composition

VM/SP consists of four components: Control Program (CP), Conversational Monitor System (CMS), Remote Spooling Communications Subsystem (RSCS), and Interactive Problem Control System (IPCS). Each of these components controls its unique part of the system. Together, these components provide the virtual machine with time sharing, remote spooling, and problem reporting for System/370 architecture uniprocessor, attached processor, and multiprocessor systems. In publications, this collective package, that is, CP, CMS, RSCS and IPCS, is referred to as VM/SP.

The components RSCS and IPCS are technically at a Release 6 level of the product. They do not contain new function to be supportive of the new function provided in CP and CMS. However, there are program product counterparts to RSCS and IPCS that have been technically advanced to work with new function provided in VM/SP.

These recommended program products are:

- Remote Spooling Communications Subsystem (RSCS) Networking, program number 5748-XP1
- Interactive Problem Control System Extension, program number 5748-SA1

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 a virtual machine. CMS provides a wide range of general-purpose, conversational time sharing functions.

REMOTE SPOOLING COMMUNICATIONS SUBSYSTEM

The Remote Spooling Communication Subsystem (RSCS)[1], a VM/370 Release 6 component, 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)[1], a VM/370 Release 6 component, 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 machine's operator's console.

---

[1]Neither the RSCS or IPCS components of VM/370 Release 6 have been functionally enhanced to be compatible with all of the functions provided in VM/SP. IBM recommends the use of the RSCS-Networking Program Product as well as the IPCS Extension Program Product. Both program products have been updated to support the functions of VM/SP.

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/SP 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.

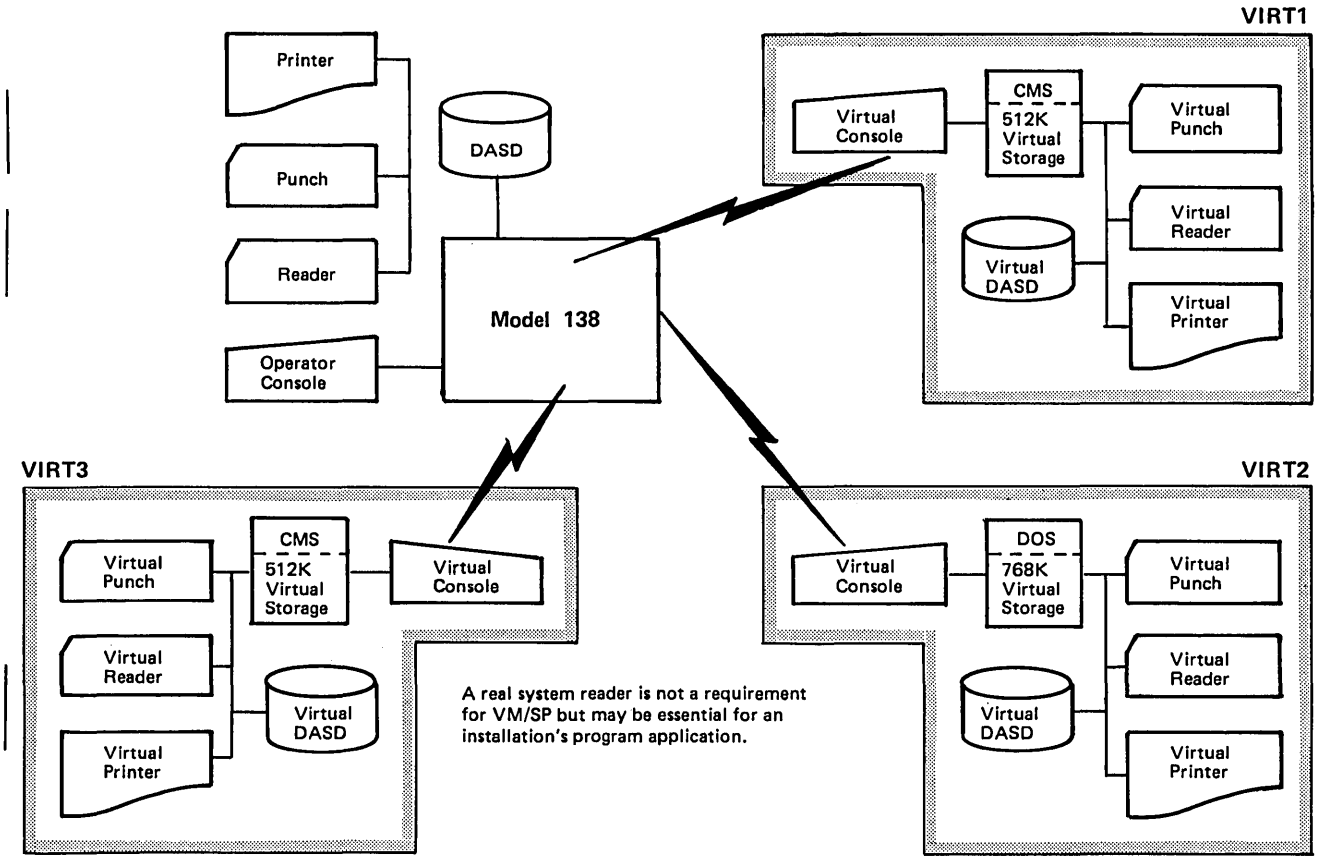


Figure 2. Simultaneous Operations of Virtual Machines

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/SP directory.

Virtual storage extends beyond the size of real storage and is not limited by the amount of real storage. It is highly probable 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 CP. Generally, one virtual machine cannot access or alter the virtual storage of another virtual machine; however, mutually consenting users may share read-only virtual storage and read-write virtual storage.

Refer to the "Control Program" section of this publication for more information on paging and storage management.

VIRTUAL PROCESSOR

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

The virtual processor provided in the virtual machine is a uniprocessor (one processor) simulation. Attached processor and multiprocessor simulation is not supported in the virtual machine environment regardless of the installation's computer complex on which VM/SP is loaded.

VIRTUAL SYSTEM CONSOLE

The virtual machine system console has three major communication functions. First, to communicate to CP (the control program portion of VM/SP) so that CP can provide to the virtual machine simulations of functions that are performed on real system consoles. Examples of this function are: IPL, START, STORE, and DISPLAY. Second, to provide the virtual machine user a means of dynamically altering specific attributes of a virtual machine; examples of attributes that may be altered are storage size, virtual machine operational control parameters and performance options. Third, to provide a means of communicating with the application program that is running in the virtual machine. To accomplish these functions requires a real terminal device. To handle simple task requirements of CP and a CMS program compile application, the virtual system may only require use of a hardcopy keyboard terminal. For more complex operations a more sophisticated device such as a multicolor display terminal may be required. For a list of the terminals VM/SP supports and their features refer to VM/SP Planning and System Generation Guide.

The list of real terminal devices supported for VM/SP virtual machine console activity is extensive. This list includes display and hardcopy devices that adhere to traditional local and teleprocessing line protocols as well as other protocols supported by System Network Architecture (SNA). SNA contains advanced communication protocols that are used with unique devices that support it. An SNA virtual machine console can only be used in VM/SP through the use of another virtual machine, a service machine, that is running an application supportive of SNA terminals. The application, VM/VTAM Communications Network Application (VCNA) Program Product, maintains a dialogue with the console services of CP through function in CP called SNA CCS (Console Communication Services). The SNA Console Communications Services (SNA CCS) provides the necessary interface between the existing CP system's console services and the VM/VTAM Communications Network Application (VM/VCNA). See the publication, VM/SP System Programmer's Guide, for more information on the SNA Console Communications Services component of CP. See the VM/VCNA publications listed in the Preface for information on using SNA terminals.

Single Console Facility

Traditionally, one real display terminal is used as a terminal console for one virtual machine. Facilities exist in VM/SP where console input and output information from multiple disconnected virtual machines can be directed to one virtual machine's real terminal.

A virtual machine is "disconnected" by detaching its assigned supporting terminal console. This is done by CP command. Note, disconnecting the console does not negate current virtual machine processing.

This facility, the Single Console Facility of VM/SP, uses directory supplied values to direct any console activity from the disconnected virtual machine to another virtual machine's console.

This facility is helpful when observations of the interactions of two critical virtual machines can best be

served by one console. In another situation, where the console I/O activity of some service machines is slight, it would be economically feasible to have their combined activity consolidated onto one user's display terminal.

Virtual I/O Devices

The virtual machine supports the same devices as a real machine: it is the virtual machine, not CP, that controls them. The I/O configuration must be defined by the user in the CP's user 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 addresses 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/SP allows the same virtual address to be used by multiple users.

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

- Logical devices (Logical Device Support Facility)
- Dedicated devices
- Virtual disks (minidisks)
- Virtual unit record devices
- Virtual transmission control units (virtual lines)
- Channel-to-channel adapters

LOGICAL DEVICES

One form of virtual device is the logical device. The logical device is a product of the Logical Device Support facility of CP. Virtual devices, as stated earlier, are generated via the user directory statements at the time of logon or afterwards by the user via CP commands; whereas, the logical device is created by CP by the execution of special code (DIAGNOSE) in a virtual machines program instruction stream.

Generally, the virtual device belongs exclusively to the virtual machine for which it was created, not so the logical device; the logical device's affiliation can be to the virtual machine that was instrumental in creating it. Or the logical device can be created for use by another virtual machine or another host system. For more information on logical device support, see the VM/SP System Programmer's Guide. The VM Pass-Through Program Product (5748-RC1) is an example of one program that uses VM/SP's Logical Device Support Facility.

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 or 370x 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 CP.

Mass Storage System

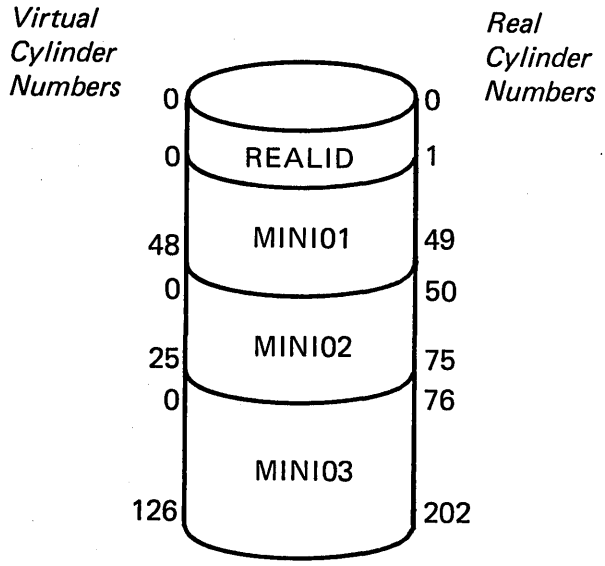
The 3850 Mass Storage System (MSS) can be dedicated to as many as four virtual machines running on VM/SP. The mass storage control (MSC) in the 3851 Mass Storage Facility (MSF) accepts requests for data from one to four System/370 processors. Each of the connections can be attached either to a different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

With MSS, there are available to the host processor as many as 192 DASD 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: CP 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/SP 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 real disk. These disks vary in size and can be used by CMS, DOS, or OS. For FBA devices, virtual disks are at least the size of one block and can be as large as the capacity of the real device. Figure 3 shows three 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/SP 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, CP 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.

Often, the user's application program may require additional DASD for its processing needs. To this end, the user can dynamically increase his virtual DASD space allocations as the need arises. This DASD space, called Temporary or T-disk storage, is acquired and relinquished from the users virtual machine as usage dictates. Information left on such disks at the end of a users session can be erased by the system prior to T-disk space reassignment. Whether or not to erase T-disk space is an installations system generation consideration (SYSCLR macro) and is dependent on the data security requirements of the virtual machines it services.

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. Also spooled output from one virtual machine can be directed to the spool input device of the same virtual machine or to another virtual machine. 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 virtual 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.

There are alternatives to a virtual channel-to-channel adapter. One is the virtual machine communication facility (VMCF); another is the Inter-User Communication Vehicle (IUCV). Both of these facilities are described briefly in this section under the topic "Other System Features."

VM/SP User Directory

Critical to the virtual machine is its definition in the VM/SP user directory. Each virtual machine is defined at system installation. The virtual machine directory entry, composed of control statements, defines the power and scope of each virtual machine. Some of the items that are defined for virtual machines are:

- The identification and password that must be used to gain access to a virtual machine
- The virtual storage size limitation
- The commands that a user may use in controlling the virtual machine (command privilege classes)
- The virtual machine's unit record devices (that is, reader, printer, punch spool devices)
- The permanent DASD assigned to the virtual machine
- The linkage requirements needed to access other virtual machines DASD
- The special options that relate to the virtual machine's performance and operational requirements
- The IUCV authorization requirements for users who require IUCV virtual machine to virtual machine dialogue.

The Directory resides on the CP system residence DASD. The Directory is accessed by CP when a prospective user attempts to gain access to a virtual machine. The process of gaining access to a virtual machine is called the logon process. If logon is successful, the user's virtual machine is built in real storage. The user can then control it through virtual console activity or other programming techniques.

Many of the virtual machine's characteristics defined in the user directory can be modified by commands. The use of commands (invoked anytime after logon) allows the temporary modification of certain virtual machine attributes for the duration of the current virtual machine session.

To change options on a more permanent basis for a user's future virtual machine activity requires intervention by administrative personnel to modify the directory and reinstall it on the system.

A more streamlined method of directory updating is handled by the VM/Directory Maintenance Program Product, (Program Number, 5748-XE4). This program allows the virtual machine user and administrative personnel to share in the user directory update process thus reducing much of the administrative overhead. For details on this product, refer to the VM/Directory Maintenance Program Product General Information Manual, GC20-1836.

LOGON PROCEDURE

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

Figure 4 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 4. Logging onto VM/SP and Loading CMS

LOGON/AUTOLOG AND LINK JOURNALING

The integrity of a user's virtual machine and DASD space is directly related to the security of VM/SP. VM/SP has an installation option that, if used, forces the masking of passwords that are used in virtual machine Logon and in the DASD access process[1]. This masking of the password prevents the inadvertent disclosure of the password to any unauthorized personnel.

VM/SP can monitor and record unsuccessful logon attempts. In addition, VM/SP can also journal all successful and unsuccessful DASD linkage attempts. If system generation selected values (thresholds) are exceeded, an alert message is sent to the VM/SP system administrator by CP.

Virtual Machine Operating Systems

While the control program of VM/SP 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 5 lists some of the operating systems that can execute in virtual machines.

```
Batch or Single-User Interactive
   DOS
   DOS/VS
   DOS/VSE
   VSE/AF
   OS/PCP
   OS/MFT
   OS/MVT
   OS/VS1
   OS/VS2
   OS-ASP
   RSCS - Networking

Multiple-Access
   VM/370
   VM/SP
   Time Sharing Option of OS

Conversational
    CMS
```

Figure 5. Virtual Machine Operating Systems

With the exception of OS/PCP and CMS, these are all multiprogramming systems. However, when operating in a virtual machine, 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/SP. 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 execute 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/SP 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.

---

[1]Exempt from installation option control are supported typewriter terminals that are not equipped with the print inhibit feature. For these devices, VM/SP types masking characters over which the password is typed during the logon process.

Multiple-Access Systems

Multiple-access systems, such as MVS/SP, 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 TSO system executing in a virtual machine. Once his terminal is connected, the user communicates only with that particular version of TSO.

OTHER PROGRAMS AND SYSTEMS

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

VM/SP Applications

Using VM/SP, 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 System Product Editor to create source programs and data files
- Using the CMS UPDATE command and the system product 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/VS, a program product, for text preparation, to create and update program specifications.

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

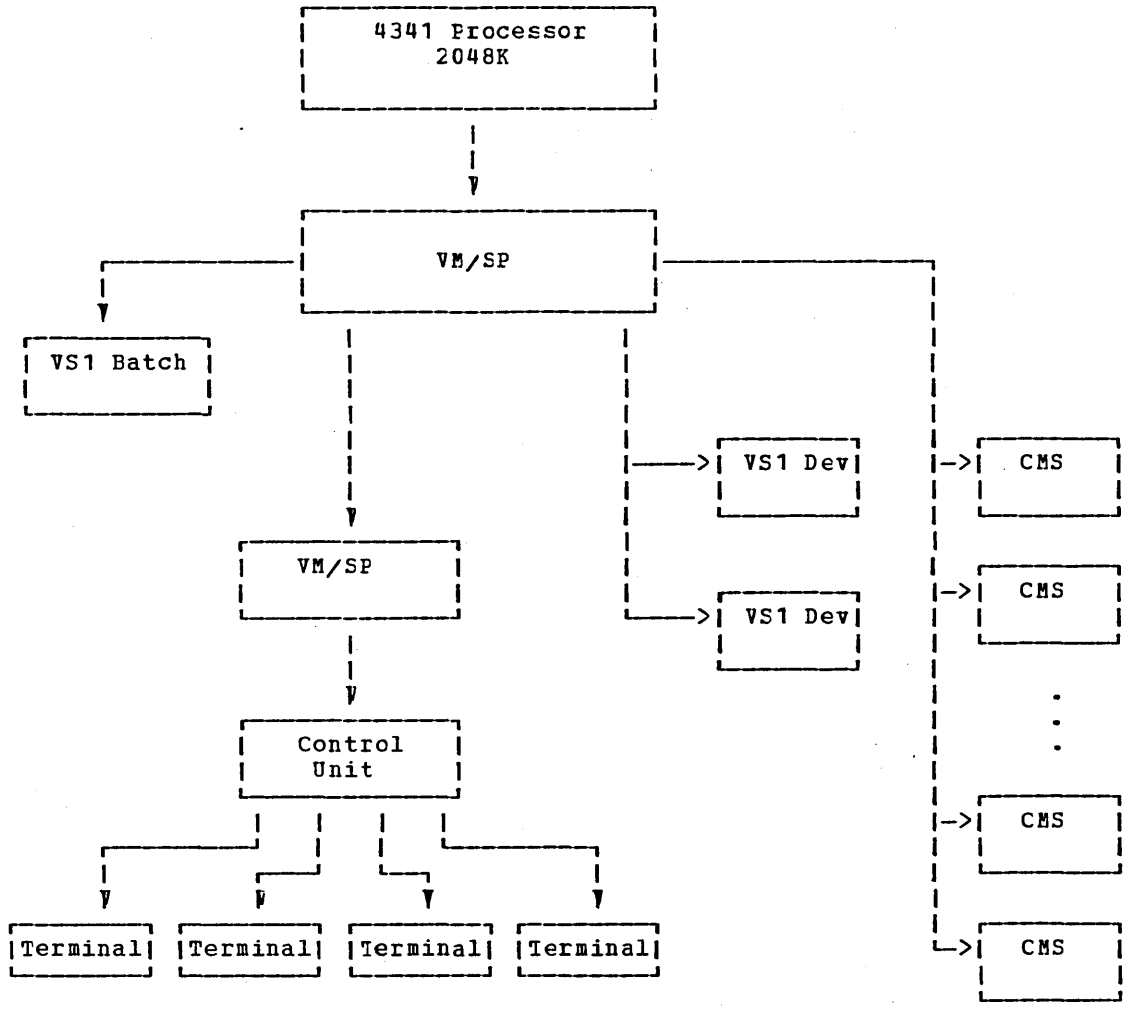


Figure 6. Virtual Machines Running Concurrent Production, Development, and Testing

- 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/SP configuration is shown in Figure 6. Operating system testing is done concurrently with batch work and a variety of conversational applications.

Backup System

An installation using VM/SP 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 and Performance Options

The performance of any computing system is judged by how efficiently it processes the work it has to do. The following factors influence the performance of a VM/SP system: the processor model used, the total

number of virtual machines executing, the type of operating systems being used in the virtual machines, the type of work being done by each virtual machine, the type, capacity, and number of primary paging devices, the number of channels available, the channel operating mode (block multiplexer or selector), the amount of real storage available, the use of alternate paths to I/O devices, and the use of Attached Processor (AP) support or multiprocessor (MP) support.

In addition to the above considerations, VM/SP offers the following performance options to enhance the operating efficiency of one or more virtual machines. The performance options are:

- Virtual machine assist
- Extended Control-Program Support (ECPS:VM/370)
- Locked pages
- Queue drop elimination
- Reserved page frames
- Virtual=real
- Priority
- Favored execution
- Affinity
- Nondisruptive Transition To and From Native Mode
- Single Processor Mode
- Shadow Table Maintenance
- VM/VS Handshaking
- S/370 Extended Support

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

VIRTUAL MACHINE ASSIST

When an operating system is run in a virtual machine 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.

EXTENDED CONTROL-PROGRAM SUPPORT

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

LOCKED PAGES

The system operator can, by use of a command, lock virtual storage pages into real storage to increase the efficiency of CP or selected virtual machines. This is done to eliminate paging activity on these pages.

QUEUE DROP ELIMINATION

The SET QDROP userid OFF command can be used to prevent a virtual machine's pages from being scanned or flushed when the virtual machine is dropped from the active queue because it is idle. Use of the SET QDROP userid OFF command can improve performance in cases where a cycle of queue dropping and reactivation is repeatedly executed.

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 allows one selected virtual machine to run totally within real storage. This eliminates all paging activity for this virtual machine (except page 0). Use of this option requires that a virtual=real area be allocated to real storage during the system generation process. In addition, virtual machines that are to use this area must have the virtual=real option specified in their user

directory option statement. Note, only one virtual machine can use the virtual=real storage allocation at any one time. The virtual=real option usually results in a performance improvement for the selected virtual machine, however, it must be understood that the dynamic paging area for all other concurrently operating virtual machines is reduced by the virtual=real area.

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 can provide a particular virtual machine an assured percentage of real processor time.

AFFINITY

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

NONDISRUPTIVE TRANSITION (NDT) TO AND FROM NATIVE MODE

Dynamic system control program (SCP) transition to or from native mode enables an operator to dynamically switch an SCP from a guest V=R virtual machine environment to a stand-alone (or native) environment and vice versa. There is no need to initial program load (IPL) the SCP or VM/SP to make the switch. After the transition of being switched to native mode, the SCP can resume operations. For details on the operation of dynamic SCP transition to and from native mode, refer to the VM/SP Operating Systems in a Virtual Machine.

SINGLE PROCESSOR MODE

Single processor mode improves the throughput of OS/VS2 MVS attached processor (AP) systems or tightly coupled multiprocessing (MP) systems running under VM/SP. To use single processor mode, MVS must be running in the V=R virtual machine.

When an MVS AP or MP system runs under VM/SP without single processor mode, MVS runs on only one processor. Running in VM/SP's single processor mode, MVS runs on two processors: MVS has exclusive use of one processor while VM/SP controls the other processor. The VM/SP side of the multiprocessor configuration has among its virtual machines, a large V=R machine dedicated to the balance of the MVS system. Single processor mode improves MVS throughput by running MVS on two processors instead of one. For details on the operation of single processor mode, refer to VM/SP Operating Systems in a Virtual Machine.

SHADOW TABLE MAINTENANCE SUPPORT

In VM/SP, CP maintains tables that relate real storage allocations (first level storage) to the virtual storage provided to the virtual machine (second level storage). The virtual machine's VS system (Extended-control mode) provides additional tables to control its own virtual storage (third level storage). CP maintains tables that map first level storage to third level storage for use by the real machine in its paging operations; these tables are called shadow tables. Maintaining shadow tables constitutes a considerable amount of overhead for CP. The Shadow Table Maintenance Support eliminates much of the overhead of shadow table maintenance in the following areas:

- Multiple shadow table support
- Selective invalidation
- Shadow table bypass for V=R users
- Shadow table bypass for V=V users

Briefly, maintaining multiple shadow tables relating to the multiple address spaces that are used by a guest operating system reduces the need of shadow table purge operations. VM/SP will dispatch the proper shadow table when the virtual machine operating system dispatches a new address space.

The selective invalidation function allows entries within the shadow tables to be selectively invalidated (rather than invalidating and purging the entire table)

when VM/SP steals or releases a page from a virtual machine.

Shadow table bypass for V=R users allows the elimination of the shadow tables and the associated CP overhead in maintaining them. This is achieved because all the virtual machine pages are mapped directly to real pages with the exception of page 0.

Shadow table bypass for the V=V user also reduces CP overhead in maintaining the associated shadow tables. The address spaces in guest operating systems may have segments or pages which are mapped V=R; also, guest systems may have common segments which are common between the different address spaces. These areas can be defined by CP command with the effect that these areas become common in the shadow tables. This reduces CP overhead in maintaining the set or sets of shadow tables for the virtual machine.

VM/VS HANDSHAKING

VM/VS handshaking is a communication path between CP and the following operating systems running as virtual machines under VM/SP:

- DOS/VS Release 34 Advanced Functions -- DOS/VS program product (5746-XE2)
- DOS/VSE with VSE/AF program product (5746-XE8)
- VS1 Release 4 and subsequent releases

To improve their operation with VM/SP, 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/SP 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 nonpaging 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/SP.

VM/SP SYSTEM/370 EXTENDED SUPPORT

VM/SP supports the System/370 extended facility of the 3031, 3032, and 3033 processors. It also supports the System/370 extended feature of the Model 158 and 168 processors. The VM/SP support allows a virtual machine to make use of the System/370 extension functions, provided the feature is installed. The functions provided by the System/370 extended feature or facility include:

- Low address protection
- Common segment support
- MVS assist operations
- Invalidate Page Table Entry (IPTE) instruction
- Test Protection (TPROT) instruction.

For more information on VM/SP support of System/370 extensions, refer to VM/SP System Programmer's Guide.

Performance Measurement and Analysis

The VM/SP 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/SP records the error in its own error recording area. Therefore, error

records from all operating systems reside in this one centralized error recording area.

Other System Features

VM/SP 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
- Inter-User Communication Vehicle

VIRTUAL MACHINE ACCOUNTING

VM/SP keeps track of a virtual machine's usage of system facilities and records accounting information whenever the use of some chargeable resource is terminated. VM/SP accounting information is spool file data. By using VM/SP system generation controls, this information can be directed to a virtual machine that specializes in accounting transactions such as billing and resource analysis. Options of punching this information via a real punch or dumping this information to tape are also available.

SAVED SYSTEMS

When initially loading an operating system into a virtual machine by device address, VM/SP 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.

In addition, a system programmer can specify certain requirements to ensure the integrity of a virtual machine and to successfully recover certain virtual machines. At system generation, the system programmer can specify that the virtual machine contents of specified users (via a system generation macro) be saved automatically on DASD if either VM/SP terminates the virtual machine or if VM/SP itself is terminated.

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 or a multiprocessor 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/SP. 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/SP 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/SP 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 uniprocessor, attached processor and multiprocessor 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/SP system. A more detailed description of VMCF can be found in the "Control Program" section of this publication.

INTER-USER COMMUNICATION VEHICLE

The inter-user communication vehicle (IUCV) defines a precise protocol for communication between virtual machines operating under the same VM/SP system. In addition, it is possible for authorized virtual machines to communicate with console communication services portion of VM/SP's control program. For more detail on the function provided by IUCV, refer to the "Control Program" section of this publication.

PROGRAMMABLE OPERATOR SUPPORT

Programmable Operator allows operation of a VM/SP system to be automated with predefined procedures, or controlled from a remote VM/SP host. For more information, see the "Control Program" section of this book.

TRACE TABLE RECORDING FACILITY

The Trace Table Recording Facility provides Field Engineering and service personnel with the ability to create a synchronized collection of trace table data. A more detailed description can be found in the "Control Program" section of this manual.

This section describes briefly the major functions provided by the control program portion of VM/SP. They are:

- Virtual Machine Time Management
- Virtual Machine Storage Management
- Virtual Machine I/O Management
- Spool Management
- Virtual Machine-to-Virtual Machine Communication Management

Communications between virtual machines are handled by Virtual Machine Communication Facility (VMCF) and the Inter-User Communication Vehicle (IUCV).

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/SP 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 user 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 user 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.

PAGING AND PAGE MIGRATION

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.

Heavily used portions of VM/SP are kept in real storage. However, to optimize real storage use, pages, both CP and virtual machines, that indicate high paging activity should be relegated to paging devices and facilities that have high access speeds; namely: the 2305 drum, DASD fixed-head area, and areas of DASD where

the access travel time (mechanical seek time) is minimized. Such areas on drum and DASD are preferred paging areas. Areas beyond these designated areas are referred to as nonpreferred areas. The allocation and grouping of preferred and nonpreferred areas are VM/SP system generation considerations.

The dynamics of CP's paging activity can saturate or fill the preferred and the nonpreferred DASD space; therefore, a page migration function of CP is used to move less active pages to nonpreferred DASD and also to migrate pages that have increased paging activity from nonpreferred DASD to preferred DASD locations.

Page migration occurs automatically when system threshold limits are attained. CP's MIGRATE command can also be used to initiate page migration.

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

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

In areas where the preferred space is heavily used and page migration is unable to identify unused pages on the preferred DASD space, the system will be switched to fair share allocation mode. When the demand lessens and the number of active pages becomes less than the available preferred DASD space, the system will switch back to normal allocation mode.

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

VIRTUAL STORAGE PRESERVATION

The system programmer can specify, at system generation time, that the virtual machine contents of certain userids are to be saved automatically if either the virtual machine is terminated by VM/SP, or if VM/SP itself is abnormally terminated. The contents of these virtual machines are saved on DASD space the VM/SP system programmer allocates. The user can restore the saved virtual machine via the IPL command and then initiate normal recovery procedures for the virtual machine. The automatically saved virtual machine is made available to only two previously specified userids. This is done in order to preserve the privacy and security of the virtual machine. The virtual machine's IPL is altered so that the saved virtual machine can be loaded into either a V=R or a normal non V=R machine. Where multiple virtual machines are to be saved the order in which to capture the contents of the VMSAVE systems can be established by system generation macro values.

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, MVS, DOS/VS, DOS/VSE, VSE/AF, VM/370, and VM/SP. 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/SP 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/SP, "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 maintains shadow pages for any virtual machine running in EC mode. A single set is all that is necessary for OS/VS1 or DOS/VS.

However, when VM/SP or VM/370 are used as a virtual machine operating system, multiple virtual storage systems can be created. In this case, the shadow tables are maintained by the host CP whenever the guest VM/SP or VM/370 passes control from one virtual machine to another.

SEGMENTS

Virtual machines can share one or more segments of virtual storage. The user has the options of having shared segments:

- Statically defined and read-only (protected),
- Statically defined and read-write

NAMED SYSTEMS

Static definition of shared segments occurs at CP system generation. Each such collection of segments is called a named system. The system programmer defines a collection of segments (their contents and keys) as a named system. This also defines whether the named system is protected (shared read-only), or unprotected (shared read-write). Thereafter users may attach or detach the named collection using a DIAGNOSE code. If the segments constitute a system control program (for example, VM/SP), the user may attach it by issuing the command "IPL system-name".

CMS is an example of a named system. The central piece of CMS is shared among users as a read-only (protected) named system which is invoked by the IPL command. Other parts of CMS (for example, support for VSAM, DOS) are separate protected named systems which are connected to virtual machines as needed via the Diagnose interface code.

A named system is attached to a virtual machine at the addresses specified when the system is defined by the systems programmer. Named systems may be attached within or beyond the extent of a virtual machine's defined storage.

When it is defined, some segments of a named system may be designated as private. Each attached user gets a private copy of such segments initialized to the initial contents of these segments. Thereafter, each user may alter his private segments of the named system. There is only one copy of the shared segments of a named system in real storage (in attached processor and multiprocessor systems there are two copies of protected shared segments). Sharing the same segments results in reduced real storage requirements and reduced paging activity. The greater the degree of sharing, the greater the real storage savings and the greater the probability the shared pages will be frequently referenced. Frequently referenced pages tend to remain in real storage thereby reducing paging activity. Reduced paging activity frees the processor for more productive work.

PROTECTED NAMED SYSTEMS

VM/SP protects the multiple users of protected named systems from any intentional or inadvertent alteration of the named system's code or data. Before CP dispatches a new user, it checks whether or not the previous virtual machine altered any shared pages of a protected named system. If so, CP places the previous virtual machine in console function mode, returns the altered page to free storage and then CP refreshes the contents of the shared page from the original disk version.

UNPROTECTED NAMED SYSTEMS

Unprotected named systems allow sharing users to alter shared pages of the named system. All users of an unprotected named system see such changes although the changes are not reflected to the disk and so do not survive CP restarts or periods in which the named system is not shared at all.

Installations choosing to run with unprotected named systems should carefully evaluate these systems to assure that they contain read-only information. The unprotected option eliminates the overhead of scanning all pages of the named system at each dispatch. In attached processor or

multiprocessor systems it eliminates the duplication of pages of the named system in real memory. For a description of shared segments see the VM/SP System Programmer's Guide.

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 user directory entry of the virtual machine, or they may be attached to (or detached from) the virtual machine while it is 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 any operating system executes in a real machine, input/output operations are initiated when a problem program requests the operating system to issue a Start I/O instruction to a specific device. Device error recovery is handled by that system. In a virtual machine, guest operating systems 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 DASD space allocation 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 resolved. 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 cylinder fault 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 to the virtual machine 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, as a general rule 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/SP 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/SP 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 user 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 actual linkage can be established.

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 where only one I/O task can be executing at one time. Another I/O task initiated to another device on the same channel will be held out with a busy condition until the first task completes. Selector channel 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 or via the user 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.

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/SP 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 machine to receive the card deck.

Input spool files can also be created by using a remote spooling work station on a network. 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 virtual machine that is controlling the spooling network 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 (or SPOOL command with the CLOSE operand) 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. The spooling operator can purge existing spool files or assign additional direct access storage space for spool files. As an alternative, the spooling operator can dump spool files to tape for later retrieval and processing. When spool space is again available, the operator can restore the dumped spool files to the system for subsequent processing. The files spooled to tape can also be hand carried and loaded into another VM/SP system for output spool file processing.

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 (or backspace the complete file), and define spooling classes and output form for real output files.

CP commands transfer and sequence (order) spool files. CP commands use identifiers imbedded in the control information associated with each closed spool file to control the ordering and transfer of the spool file. Spool file identifiers are:

spoolid - A unique number (assigned by CP sequentially from 1 through 9900) that the system assigns when the spool file is created. The VM/SP control program prevents the reassignment of duplicate spool file identifiers.

spooling class - By user directory statement or by CP command, the user can assign a one character identifier to file groupings. These assignments can represent security categories, geographic regions, or priority classes.

form number - A form number can be assigned by CP command for real spool output files. The form number is used in alerting the systems spooling operator of the output forms or card requirements of the forthcoming output spool file.

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 console is a display device or 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.

In diagnostic applications, spool console output is beneficial in the reconstruction of events that preceded a real or virtual system abend.

Remote Spooling

CP, in conjunction with remote spooling facilities, can transmit files across a teleprocessing network. The "Remote Spooling Communications" section later on in this book describes how this is done 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/SP, and by users to control virtual machines and their operating systems.

Normally, 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/SP'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/SP 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 7 for a concise description of each privilege class.

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/SP 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/SP.
B[1]	System Resource Operator	To control all the VM/SP 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/SP storage area.
F[1]	Service Representative	To obtain and examine, in detail, certain data concerning the I/O devices connected to the VM/SP system.
G[2]	General User	Control functions associated with the execution of the users virtual machine.
Any[2]	Any User	Commonly used commands that are available to all users of VM/SP for a variety of uses.
H	Reserved	Reserved for IBM use only.

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

Figure 7. CP Privilege Class Descriptions

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/SP CP Command Reference for General Users.

OTHER USERS

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

VM/SP Attached Processor and Multiprocessor Support

ATTACHED PROCESSOR SUPPORT

On systems that are either an attached processor complex or an asymmetric multiprocessor complex, VM/SP 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/SP 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 I/O 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.

MULTIPROCESSOR SUPPORT

VM/SP multiprocessor support functions in a manner that is similar to attached processor support except that I/O can execute on both processors simultaneously. Because VM/SP can be initialized by either processor, I/O paths should be provided to the system residence device and critical CP-owned DASD from both systems in case one system's channels or processor has to abort operations. Establishing parallel paths to critical I/O devices allows automatic system recovery on the remaining system configuration. Depending on the magnitude of the malfunction, recovery can result in the system's continuing operations in multiprocessor mode with one or more channels marked offline.

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/SP 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/SP System Programmer's Guide.

Inter-User Communication Vehicle

The Inter-User Communication Vehicle (IUCV) defines a precise communication protocol for dialogue between virtual machines operating under the same VM/SP system.

IUCV communications are accomplished by the use of a special privileged instruction operation code. Up to 32,767 bytes of information can be transferred in a single IUCV operation. The information is moved from the sender's virtual storage to the receiver's virtual storage.

Other features include:

- User directory entries to control IUCV authorization and use
- Multiple communication paths between virtual machines
- User control to selectively enable and disable IUCV interrupts for priority and nonpriority messages, for priority and nonpriority replies, and for IUCV control interrupts
- User ability to receive messages and replies either synchronously or asynchronously (via external interrupts)
- Tracking IUCV function in the CP trace table
- Communicating with Console Communication Services (CCS), a function of VM/SP's control program, by authorized virtual machines

For details on the functions performed by IUCV, refer to the VM/SP System Programmer's Guide.

PROGRAMMABLE OPERATOR SUPPORT

The Programmable Operator facility can be used in a VM/SP system to reduce or eliminate system operator message activity. It provides a capability to:

- log messages
- suppress messages
- redirect messages
- execute commands
- preprogram message responses

under control of a changable message routing table.

Programmable Operator can be used by a stand-alone VM/SP system or be controlled by an operator at a host VM/SP system. Remote operation of a VM/SP system requires Release 2 or 3 of the RSCS Networking program product on both of the VM/SP systems. Release 3 of the RSCS Networking program product provides improved display usage.

TRACE TABLE RECORDING FACILITY

The Trace Table Recording Facility preserves selected trace table records plus CP and virtual machine interface records by recording them on a CP spool file. The facility, invoked by the command CPTRAP, is intended for the analysis of VM/SP problems that escape detection using a system dump.

The CPTRAP command creates a READER spool file. A CMS data reduction program, TRAPRED, is provided to allow use of the data collected. Using the spool file as input, output can be either a spooled print file or an interactive terminal display. This allows:

- selectivity of output
- positioning within the file
- displaying data in a forward or backward direction
- printing
- stopping

CP servicability is enhanced in that CPTRAP support allows Field Engineering to collect problem determination information via a telephone conversation with site personnel, who can be told how to set up a CPTRAP run before recreating the problem.

For detailed information concerning the use of the Trace Table Recording Facility, refer to the VM/SP System Programmer's Guide.

The conversational monitor system (CMS) is a major component of VM/SP. Together with the control program of VM/SP, it provides a time-sharing system suitable for direct problem solving and program development. CMS is an operating system that executes in a VM/SP virtual machine. (CMS uses the Diagnose interface for all of its disk and tape input/output operations and relies on CP's error recovery routines.)

The user can create, update, and manipulate files as well as compile, test, and execute problem programs. These interactive capabilities are extended to DOS users via the CMS/DOS environment of 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 256K bytes of virtual storage. 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.

Additional portions of CMS can be shared in discontiguous segments. Discontiguous segments can be automatically 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 and disk 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. CMS disks must be assigned in units of full cylinders for CKD DASD or blocks for FBA devices.

Figure 8 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 that are supported by CP (with the exception of the 2305). 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, but not write, DOS sequential, OS sequential, and OS partitioned datasets. CMS can read and write VSAM datasets.

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

Except for the EXECIO command, 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 labeled and unlabeled tapes via CMS commands. For more details on this, see the heading: "CMS Tape Label Processing."

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.

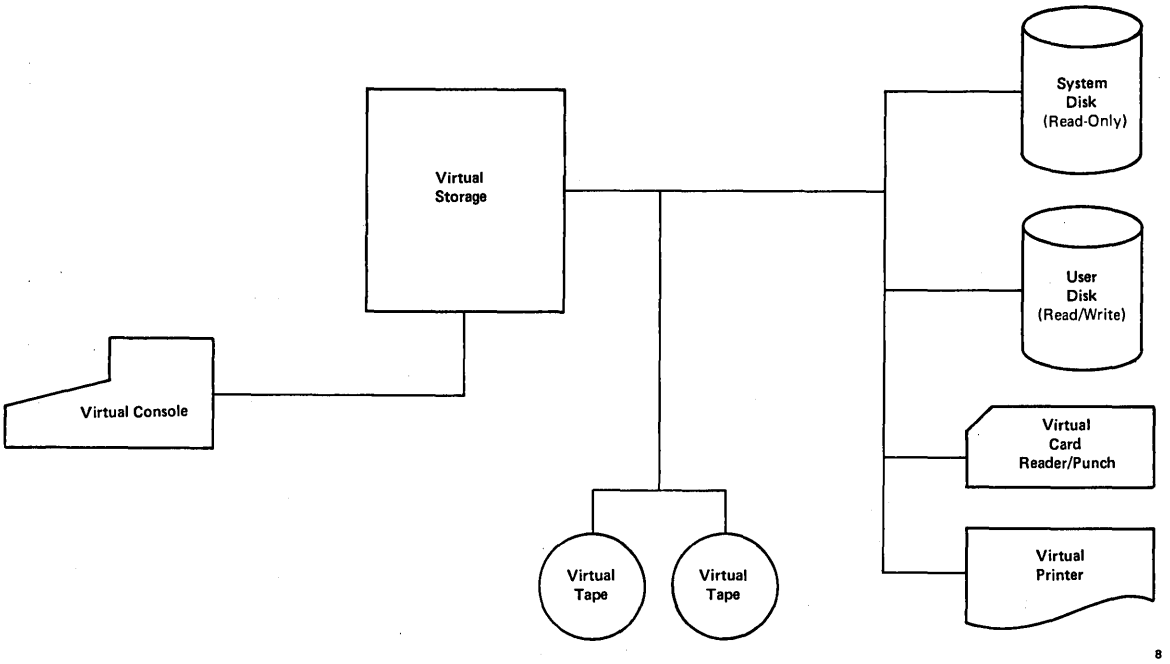


Figure 8. Sample CMS Configuration

USER DISKS

CMS requires the system residence volume to be online. In addition to the system disk, each user may have up to 25 virtual disks online at any one time. (All 25 of these disks can reside on one real disk.)

The user disks are differentiated by a filemode letter designation, assigned when the disk is made active. The filemode letter (A-Z) defines a standard order-of-search for disk files. The standard order-of-search is sequential from A through Z. The search order can be modified by a CMS command. The user's knowledge of the arrangement of files on CMS disks coupled with a search order modification can significantly reduce the time required to locate the desired files. If duplicate file identifiers exist for different files on different disks, and the files that are to be retrieved are located on a disk further on in the search order, then altering the search order is the procedure for retrieval.

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) -- identifies the disk that contains the file and the access mode assigned to the file.

```
SAMPLE  ASSEMBLE  An
```

Figure 9. CMS File Identification

Figure 9 is a sample CMS file identifier 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 n is the filemode number (the default value is 1).

One method of file creation is a result of using CMS commands, for example: XEDIT or COPYFILE and specifying the filename, filetype, filemode of the file being generated. Other CMS files are created and given file default identifications by the CMS command process itself.

FILE STRUCTURE

CMS disk files are written as fixed-length records, which usually are not physically contiguous on the disk. They are allocated and deallocated automatically by CMS as the file size demands. The user can elect 1024, 2048, or 4096 byte record blocks as a substitute for an 800 byte block allocation. 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 a chain of records that are the actual files.

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). CMS disk files that are written as 800-byte records are 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. The 3310 and 3370 disk storage devices are not supported by the old (800-byte) CMS file system.

For 1024-byte, 2048-byte, or 4096-byte record disks, a single user file is limited to a maximum of $2^{31}-1$ items and must reside on one virtual disk. The file management system limits the number of files on any one virtual disk to $2^{31}-1$.

Initialization and Dump Restore

The Device Support Facilities Program (5747-DS1) standalone version is used in a virtual machine or in a standalone manner to initialize Count Key Data (CKD) or Fixed Block Architecture (FBA) formatted DASD or minidisks for OS or DOS. Alternatively, a guest SCP may be used to initialize CKD DASD.

The CMS FORMAT command initializes minidisks for CMS.

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

The DASD Dump Restore (DDR) program of VM/SP, which executes standalone or under CMS, dumps, restores, and displays all types of real disks and 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.

In addition to the above, CMS provides as an aid to all users, a HELP command facility. This facility allows users to display pertinent information on all CMS commands, all CP General User Commands, and system messages. The CMS HELP facility has function that permits users to tailor this text data base by the addition and subtraction of information to fit the installation's user population needs. The user can modify this data base to include commands and programs of his own creation.

The EXEC processors of CMS, that is CMS EXEC and EXEC 2, 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 processors have logical capabilities; EXEC procedures can test the contents of variables, branch on specified conditions, and execute programmed loops. A special EXEC procedure, named: PROFILE EXEC can be invoked automatically when the user issues his first command in the CMS environment; information in the PROFILE EXEC can contain CP and CMS commands that upon automatic execution tailor the virtual machine to meet the user's immediate terminal session requirements.

The EXEC 2 processor handles EXEC 2 programs. EXEC 2 programs and EXEC 2 processing are similar to CMS EXEC programs and CMS EXEC processing. The EXEC 2 processor provides advanced EXEC programming capabilities and improved performance for the CMS user.

You may not use CMS EXEC language statements in a program to be interpreted

by the EXEC 2 processor, nor EXEC 2 language statements in a program to be interpreted by the CMS EXEC processor. However, you may call an EXEC 2 program from a CMS EXEC program, and vice versa.

Since both CMS EXEC and EXEC 2 files are called in the same way, CMS examines the first statement of the EXEC file to determine which EXEC processor must handle it. If the first statement of the EXEC is &TRACE, CMS calls the EXEC 2 processor to handle it. If the first statement is not &TRACE, CMS calls the EXEC processor to handle it.

See the publication VM/SP EXEC 2 Reference for further information about EXEC 2. See the publication VM/SP CMS User's Guide for further information about CMS EXEC.

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 (normally, the A1 disk). They are identified by the input filename together with the filetype TEXT or LISTING.

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 DOS/VS COBOL user specifies the compiler options on an OPTION command which precedes the FCOBOL command. Compiler options can also be specified using the CBL control statement in the COBOL program. 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 (with a file type of DOSLIB) and accepts most of the DOS linkage editor control statements as input.

LANGUAGE PROCESSORS

A VM/370 assembler is distributed as part of the VM/SP 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, OS FORTRAN, and VS FORTRAN are useful languages for problem-solving applications. DOS/VS, OS/VS COBOL, DOS and OS PL/I, DOS/VS RPG II, and assembler language programs assembled with the VM/SP assembler 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 or DOS 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 on that System/370 in VSE or OS virtual machines under VM/SP.

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

In an attached processor (AP) or multiprocessor (MP) 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 EDITORS

CMS contains two editors; one is referred to as the CMS Editor, the other, the VM/System Product 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. The CMS Editor is a line editor; that is, the target for editing is one line of text at a time.

The System Product Editor consists of the XEDIT command and its subcommands and macros. It is both a full screen and a line (or typewriter) editor. The VM/SP editor provides expanded function that will allow, in addition to most of the capabilities of the CMS editor:

- Multiple views of the same file or different files on the display screen
- The issuance of selected commands from the displayed line
- Provides language expansion through the EXEC 2 processor
- Plus other function

Both editors, however, remain available in order to facilitate migration. See the publications VM/SP System Product Editor User's Guide and the VM/SP System Product Editor Command and Macro Reference for further information about the System Product Editor. Details on the CMS editor is contained in the VM/SP CMS User's Guide and in the VM/SP CMS Command and Macro Reference.

SCRIPT/VS

SCRIPT/VS is a text processor available as a program product for use under CMS. SCRIPT/VS 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 one of the EXEC processors of CMS.

CMS TAPE LABEL PROCESSING

Tape label processing exists only in CMS and is not used by the CP function that provides tape output. CMS tape label processing includes the following:

- Routines to check IBM standard labels on input
- Routines to write IBM standard labels on output
- The facility to specify user exits for processing nonstandard labeled tapes during execution of CMS macro simulations and some CMS tape operation commands
- The facility to specify user-written routines to process standard user labels during DOS and OS macro simulation under CMS
- A modified CMS tape command that displays the VOL1 label on read commands and writes a new VOL1 label on write operations
- A facility to bypass label processing or to process tapes without labels

CMS supports the processing of IBM standard tape labels for OS simulation, CMS/DCS, CMS commands and the TAPESL macro. It also supports user written nonstandard tape label processing routines for OS simulation and CMS/DOS.

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. An 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/SP programming facility that executes under CMS. It allows a VM/SP 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.

For details on using CMS Batch, refer to the VM/SP CMS User's Guide.

Telecommunication support to transfer files between remote stations and virtual machines using binary synchronous (BSC) switched or nonswitched lines can be accomplished by either of the following programs:

- Remote Spooling Communications Subsystem (RSCS), a component of VM/370 Release 6
- Remote Spooling Communications Subsystem (RSCS) Networking Program Product (5748-XP1)

Both programs contain similar function; however, the RSCS component of VM/370 has not been updated for VM/SP. It does not utilize any of the VM/SP enhancements that have been added since VM/370 Release 6.

The RSCS component of VM/SP cannot:

- handle virtual 3800 files
- use a system disk formatted in 1K, 2K or 4K block size
- use a system disk on a 3375, 3380 or FBA device
- use a dedicated remote 3270 information display printer

RSCS-Networking is the recommended product to use with VM/SP.

Either of these programs using the spool file system of CP, provides a telecommunication environment for the transfer of files from:

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

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

Both of these programs have a supervisor and line drivers. The supervisor is an interface between the CP spool system and the program's line driver. The line drivers drive, or control, a specific type of remote station.

Figure 10 shows the relationship among the virtual machine users, the CP spool system, and the remote stations.

Remote station equipment consists of supported card and printer devices as well as processor configurations. Remote stations as defined for use with either RSCS or RSCS-Networking includes both programmable and nonprogrammable types.

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.

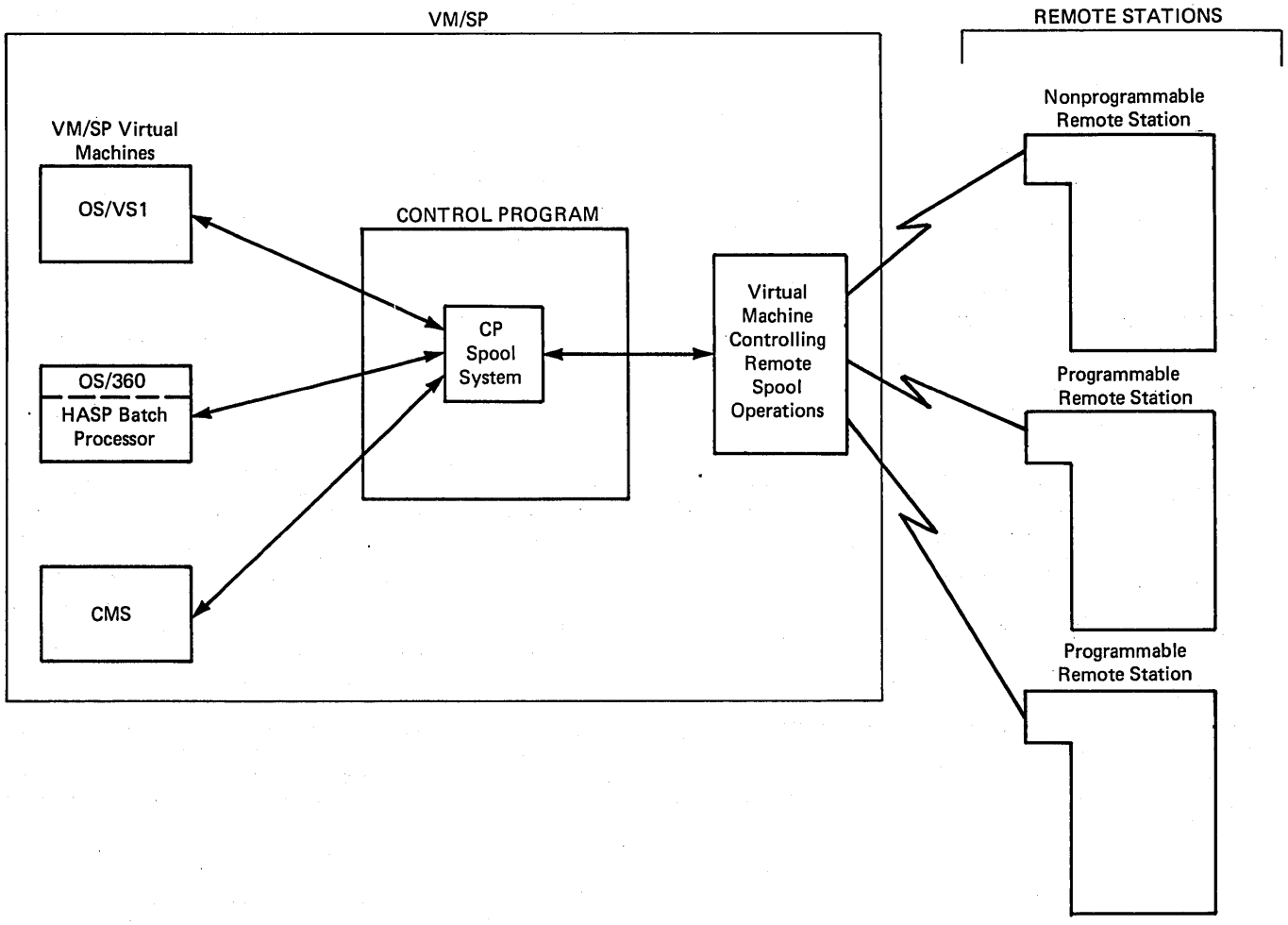


Figure 10. Relationship Among Virtual Machines, Remote Stations and CP's Spool System

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.

For 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.

For information on the RSCS component of VM/370, Release 6, refer to the RSCS User's Guide, GC20-1816.

The Interactive Problem Control System (IPCS) Extension Program Product (5748-SA1) can be used to standardize the VM/SP problem reporting process. This production enhances the serviceability of VM/SP by identifying recurring programming problems located on the 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 IPCS Extension commands. These commands are used to generate a disk-resident report. This report, along with the early warning system (EWS) microfiche, gives speedier identification of previously reported problems.

The use of IPCS Extension:

- 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.

Function is included with IPCS Extension program product to be compatible with multiprocessor support provided in VM/SP. Therefore, the IPCS program product is the suggested problem report generation program to be used with VM/SP.

For more information on this product, refer to the IPCS Extension Program Product General Information Manual, GC34-2019 and the IPCS Extension User's Guide and Reference, SC34-2020.

Similar problem reporting is provided by the IPCS component of VM/370 Release 6. However, this component is not recommended for use with VM/SP as it has not been upgraded to perform with the unique functions of VM/SP, nor does the IPCS component contain all the function provided by the IPCS Extension program product.

The following is the list of what the IPCS component of VM/370 should not be used for in a VM/SP environment:

1. Analysis, formatting, and printing of CP dumps taken when running in MP mode.
2. Analysis of CP dumps with an abend code added since VM/370 Release 6.

In addition, the user should be aware that control blocks will be formatted in Release 6 format (except for the RECBLOK which will not be formatted at all). If a block has been extended since Release 6, the extension will not be formatted.

For information regarding the IPCS component of VM/370, Release 6, refer to VM/370 Interactive Problem Control System (IPCS) User's Guide. For information on the installation of the IPCS component of VM/370, see the VM/SP Planning and System Generation, SC19-6201.

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

The following illustrates the relationship of the VM/SP audience to appropriate publications. Also included is a brief description of the text content.

Audiences of VM/SP may extend beyond the categories listed in the following chart. Such users should consult the VM/SP Glossary and Master Index, GC19-6207, to locate the topics and the related books that suit their needs.

Audience	Publications	Content
All users	VM/SP Introduction, GC19-6200	Overview of VM/SP and virtual machines
	VM/SP CP Command Reference for General Users, SC19-6211	G and "Any" commands -- needed for virtual machine control
	VM/SP System Messages and Codes, SC19-6204	Abend codes, wait states, and messages
	VM/SP Terminal User's Guide, GC19-6206	Logon/Logoff, virtual, console terminal use, attention handling
	VM/SP Quick Guide for Users, SX20-4400	Reference for command and service program control statement formats
	CMS Primer, SC24-5236	Introductory description of VM/SP CMS.
System Analyst, Programming System Representatives, Installation Planners and Support Personnel	VM/SP General Information Manual, GC20-1838	Synopsis of VM/SP, brief descriptions of unique VM/SP function, planning installation and distribution information.
	VM/SP Planning and System Generation Guide, SC19-6201	VM/SP installation requirements and procedures
	VM/SP Operator's Guide, SC19-6202	VM/SP system control commands, resources, control, service programs, spooling
	VM/SP System Programmer's Guide, SC19-6203	Resources for problem solving, tracking, tracing, monitoring
	VM/SP Operating Systems in a Virtual Machine, GC19-6212	Relationship of VM/SP with other operating systems, VM/SP concepts
	VM/SP OLTSEP and Error Recording Guide, SC19-6205	CLTS from a virtual environment. RMS, invoking CPEREP

Audience	Publications	Content
System Analyst, Programming System Representatives, Installation planners and Support Personnel (cont.)	VM/SP CMS Command and Macro Reference, SC19-6209	CMS facilities for generating, compiling editing, testing and debugging programs
	VM/SP CMS User's Guide, SC19-6210	
	OS/VS, DOS/VSE, VM/370 Environmental Recording Editing and Printing, GC28-0772	Provides details on OS/VS, DOS/VSE, VM/370 and VM/SP EREP operands invoked by CPEREP
	VM/370 Interactive Problem Control System (IPCS) User's Guide, GC20-1823	Catalog and update facility for recorded system and virtual machine problems
	VM/SP Service Routines Program Logic, LY20-0890	VM/SP logic on CP, CMS, IPCS and RSCS; the service programs; EREP and the data areas and control blocks that relate to those components
	VM/SP Data Areas and Control Block Logic, LY20-0891	
	VM/SP System Logic and Problem Determination, Vol 1 - CP, LY20-0892 Vol 2 - CMS, LY20-0893	
	VM/370 System Logic and Problem Determination, Vol 3 - RSCS, SY20-0888	
	OS/VS, DOS/VSE, VM/370 Environmental Recording Editing and Printing (EREP) Program Logic, SY28-0773	
Field Engineer (CE)	VM/SP OLTSEP and Error Recording Guide, SC19-6205	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, VM/370 and VM/SP EREP operands invoked by CPEREP
System Operator	VM/SP Operator's Guide, SC19-6202	Resource and system control command and service programs; VM/SP initialization and shutdown
	VM/370 Remote Spooling Communications Subsystem (RSCS) User's Guide, GC20-1816	Operational control of remote spooling
	VM/SP CP Command Reference for General Users, SC19-6211	G and Any commands — required for virtual machine control

Audience	Publications	Content
System Operator (cont.)	VM/SP Operating Systems in a Virtual Machine, GC19-6212	Relationship of VM/SP with other operating systems
	VM/370 Interactive Problem Control System (IPCS) User's Guide, GC20-1823	Tracking facility for system and virtual machine problems
	VM/SP OLTSEP and Error Recording Guide, SC19-6205	How to invoke CPEREP to capture/clear SYS1.LOGREC and VM/SP error cylinders
Application Programmer, Interactive users -- Anyone who needs to create or modify programs or data	VM/SP CMS Command and Macro Reference, SC19-6209	CMS command formats for program creation, editing testing and debugging
	VM/SP CMS User's Guide, SC19-6210	Detailed use of CMS CMS commands
	VM/SP EXEC 2 Reference, SC24-5219	Detailed use of the EXEC 2 processor
	VM/SP System Product Editor User's Guide, SC24-5220	Tutorial information for using the System Product Editor
	VM/SP System Product Editor Command and Macro Reference, GC24-5221	Detailed use of the System Product Editor Commands

Appendix B: Related Publications for CMS Users and Other Applications

This appendix attempts to alert users of some of the publications that support program products that run in CMS or other virtual machines. These program products provide supportive services for CP or CMS. Keep in mind that this list contains only a few of the supportive products and a still smaller list of related publications.

Since titles change and new publications are constantly being added to the IBM library, this list only serves as a guide to what is currently available. Another source for documents that relate to program products is the IBM System/370 and 4300 Processors Bibliography, Order No. GC20-0001.

As CMS is a basic requirement for the use of many of the other program products, its document requirements (including corequisite publications) are listed first.

VM/SP Quick Guide for Users	SX20-4400
CMS for Programmers, A Primer	SR20-4438
CMS Primer	SC24-5236
VM/SP CMS Command and Macro Reference	SC19-6209
VM/SP CMS User's Guide	SC19-6210
VM/SP EXEC 2 Reference	SC24-5219
VM/SP System Product Editor User's Guide	SC24-5220
VM/SP System Product Editor Command and Macro Reference	SC24-5221

Corequisite Publications

VM/SP Introduction	GC19-6200
VM/SP System Messages	SC19-6204
VM/SP Terminal User's Guide	GC19-6206

Note: In some cases, the titles are abbreviated to save space.

VS BASIC

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: Quick Guide for CMS Users	SX28-6386
VS BASIC: Installation Reference Material	SC28-8309
VS BASIC: Language Reference Manual	GC28-8303

BASIC Subroutine

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

OS/VS and VM/370 Assembler Programmer's Guide	GC33-4021
OS/VS, DOS/VS, and VM/370 Assembler Language	GC33-4010
VM/SP: System Programmer's Guide	SC19-6203

SCRIPT/VS

Document Composition Facility: Generalized Markup Language User's Guide	SH20-9160
Document Composition Facility: User's Guide	SH20-9161

FORTRAN

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
IBM OS Code and Go FORTRAN and FORTRAN IV (G1) Programmer's Guide	SC28-6853
FORTRAN IV (G1) Processor and TSC FORTRAN Prompter for OS and VM/370 (CMS): Installation Reference Material	SC28-6856

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
VS FORTRAN App. Programming: Source-Time Reference Sum.	SX26-3731
IBM FORTRAN Interactive Debug for OS (TSO) and VM/370 (CMS) Reference Card	SX28-8193
FORTRAN IV Language	GC28-6515
VS FORTRAN App. Programming: Guide	SC26-3985
VS FORTRAN App. Programming: Language Reference	SC26-3986
VS FORTRAN Installation and Customization	SC26-3987
VS FORTRAN App. Programming: System Services Reference Supplement	SC26-3988
VS FORTRAN App. Programming: Library Reference	SC26-3989
VS FORTRAN Compiler and Library: Diagnosis	SC26-3990

COBOL

Note: "IBM OS Full American Naticnal 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 CCBOL 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
IBM VS COBOL for DOS/VSE	GC26-3998

VSE VSAM and CMS VSAM

VSE/Advanced Functions Macro User's Guide	SC24-5210
VSE/Advanced Functions Macro Reference	SC24-5211
VSE/System Data Management Concepts	SC24-5209
VSE/VSAM Programmer's Reference	SC24-5145
VSE/VSAM Documentation Subset	SC24-5191
Using VSE/VSAM Commands and Macros	GC33-5382

OS/VS VSAM

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

OS and DOS PL/I Opt. Compilers: Extended Graphic Character Set Support Supplement	SC26-3971
OS PL/I Optimizing Compiler, Programmer's Guide	SC33-0006
OS PL/I Optimizing Compiler, Messages	SC33-0027
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 Cptimizing Compiler General Information GC33-0001

OS PL/I Checkout Compiler: General Information GC33-0003

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, CMS User's Guide SC33-0047

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

IPCS Extension

VM/SP IPCS Extension General Information Guide GC34-2019

VM/SP IPCS Extension User's Guide and Reference SC34-2020

VM/Interactive File Sharing

VM/Interactive File Sharing General Information Guide GC24-5195

VM/Interactive File Sharing Guide and Reference SC24-5196

VM/Pass-Through Facility

VM/Pass-Through Facility General Information Guide GC24-5206

VM/Pass-Through Facility Guide and Reference SC24-5208

VM/370 Display Management System for CMS

VM/370 Display Management System for CMS General Information Guide GC24-5197

VM/370 Display Management System for CMS Guide and Reference SC24-5198

VM/SP RSCS Networking

VM/SP Remote Spooling Communications Subsystem Networking General Information Guide GH24-5004

VM/SP Remote Spooling Communications Subsystem Networking Program Reference and Operations Guide SH24-5005

CMS LKED Command Support

OS/VS Linkage Editor and Loader GC26-3813

VM/Directory Maintenance

VM/370 Directory Maintenance
- Guide for General Users SC20-1839
- Installation and System Administration Guide SC20-1840

DOS/VS RPG II

DOS/VS RPG II
- Installation Reference SC33-6122
- Language SC33-6031
- User's Guide SC33-6074

DOS/VS Sort/Merge

DOS/VS Sort/Merge Version 2
- Installation Reference SC33-4045
- Programmer's Guide SC33-4044

GC19-6200-1

IBM VM/SP Introduction (File No. S370/4300-20) . Printed in U.S.A. GC19-6200-1

IBM Virtual Machine/System Product:
Introduction
Order No. GC19-6200-1

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language; English is not required.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

	Yes	*No*
• Does the publication meet your needs?	☐	☐
• Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

• What is your occupation? _____

• How do you use this publication:

As an introduction to the subject?	☐	As an instructor in class?	☐
For advanced knowledge of the subject?	☐	As a student in class?	☐
To learn about operating procedures?	☐	As a reference manual?	☐

Your comments:

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

If you would like a reply, please supply your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

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 NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department G60
P. O. Box 6
Endicott, New York 13760

Fold | Fold

If you would like a reply, *please print:*

Your Name _____

Company Name _____ *Department* _____

Street Address _____

City _____

State _____ *Zip Code* _____

IBM Branch Office serving you _____

IBM®

IBM VM/SP Introduction (File No. S370/4300-20). Printed in U.S.A. GC19-6200-1

IBM VM/SP Introduction (File No. S370/4300-20) Printed in U.S.A. GC19-6200-1

GC19-6200-1