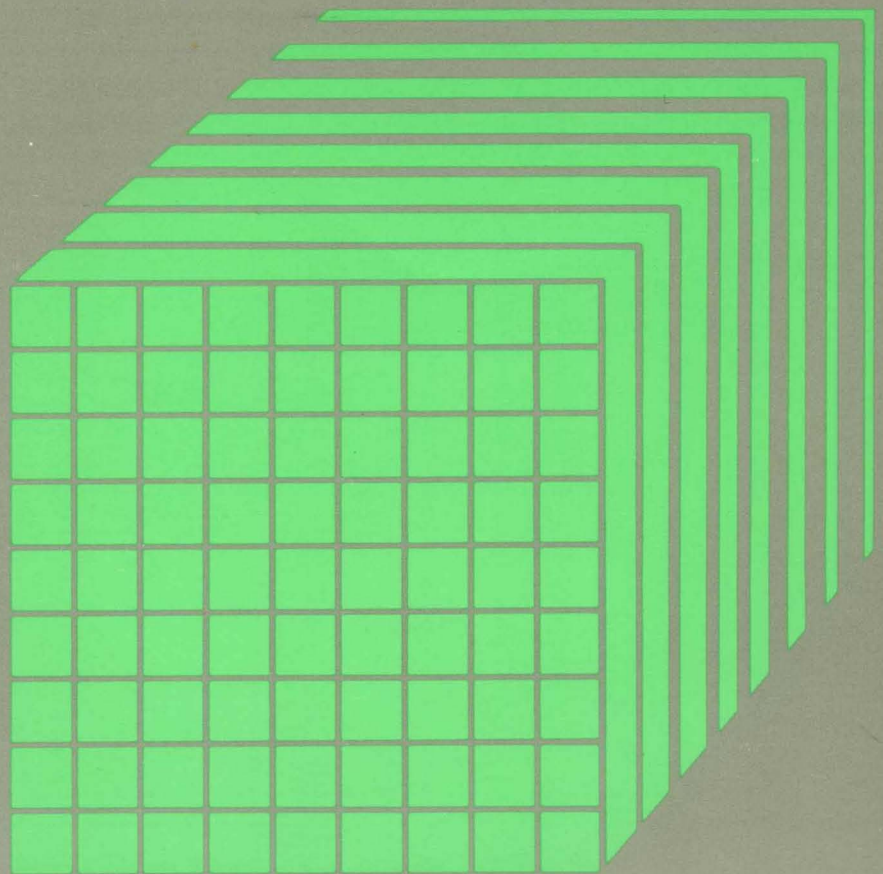




Virtual Machine/
System Product

**Operating Systems in
a Virtual Machine**

Release 3

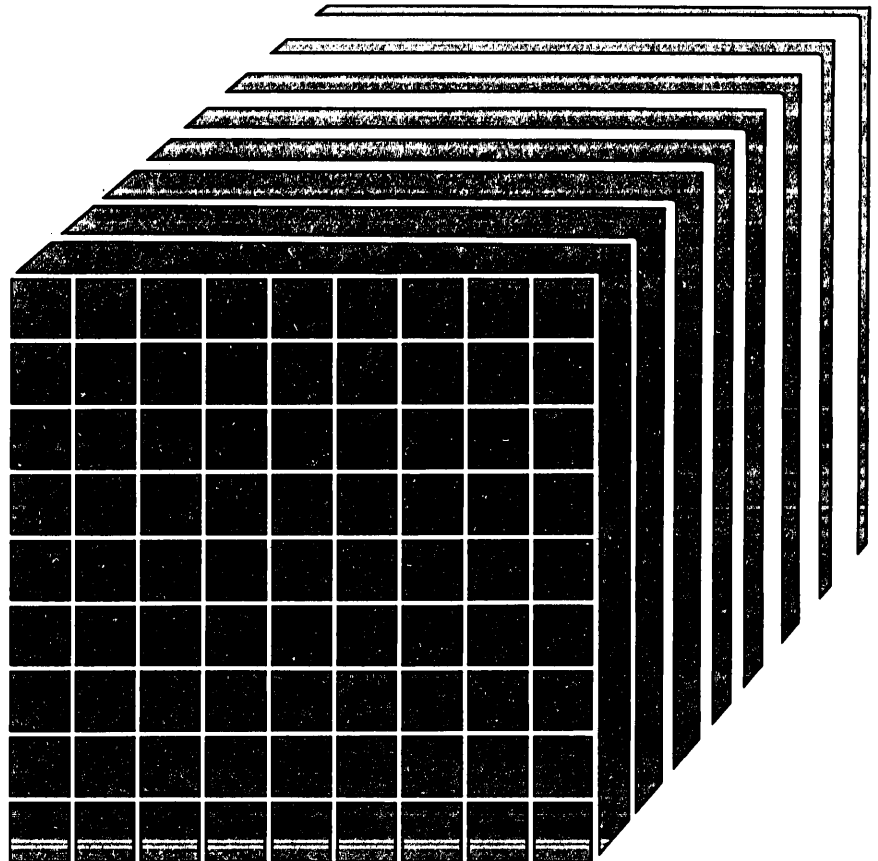




Virtual Machine/ System Product

Operating Systems in a Virtual Machine

Release 3



Third Edition (September 1983)

This edition, GC19-6212-2, is a major revision of GC19-6212-1, and applies to Release 3 of the Virtual Machine/System Product, program number 5664-167, (VM/SP) and to all subsequent releases and modifications until otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 and 4300 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

Summary of Changes

For a list of changes, see page iii.

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

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

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 readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Programming Publications, Dept. G60, P.O. Box 6, Endicott, NY, U.S.A. 13760. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Summary of Changes

Summary of Changes to GC19-6212-2 for VM/SP Release 3

Programmable Operator Facility

New: With the help of the Programmable Operator Facility, an installation can now use one operator to control all of the guest operating systems.

Program Event Recording (PER)

New: Problem determination is greatly extended by the new CP command PER.

3081 Processor

Note: VM/SP Release 3 will support AP/MP and 3081 processor Unit Model D16 in the first quarter of 1984.

Virtual Machine Options

Changed: "Section 1. General Considerations" has undergone a minor structural change so that all Virtual Machine Options are now under one heading.

3088 Multi-system Channel Communication Unit

New: The 3088 Multi-system Channel Communication Unit interconnects multiple systems. It is fully compatible with existing channel-to-channel adapters when attached to a blocked multiplexer channel,

Generic Terms

New: This publication now uses the generic terms 'DOS', 'OS', 'VSE,' 'POWER,' 'VSE/POWER,' and 'OS/VS' throughout the book. (See the Preface for a complete listing.)

Miscellaneous

Changed: Technical and editorial changes have been made throughout this publication to improve the accuracy, clarity and usability. Give special attention to the old examples that have been updated and the new examples that are included in this edition.

Summary of Changes to GC19-6212-1 for VM/SP Release 2

Device Support Facility

New: The Device Support Facility (program number 5747-DS1) replaces the service routines IBCDADDI, SURFANAL and INITDISK.

CP Serviceability

New: Problem determination capability is provided to service personnel and system programmers with the Trace Table Recording Facility. The facility uses the CP command CPTRAP. A CMS utility program, TRAPRED, is included as part of the CPTRAP facility.

3081 Processor

Note: VM/SP Release 2 will support AP/MP and 3081 Processor Unit Model D16 in the first quarter 1983.

CP Dial Support for Remote 3270 Terminal

Changed: VM/SP Release 2 allows remote 3270 terminals to use the CP DIAL command to be logically connected to a multi-access operating system.

Miscellaneous

Changed: Various minor technical and editorial changes have been made throughout this publication to promote accuracy, clarity, and usability.

**Summary of Changes
for GC19-6212-0
as Updated by GN25-0820
for VM/SP Service Level 101**

3081 Processor Support

New: VM/SP now supports the new IBM 3081 processor complex. Each 3081 processor complex includes a 3081 processor and 3082 processor controller. The monitoring and service support facility (MSSF) is an integral part of the processor controller. The support provided allows VM/SP to communicate with the MSSF to obtain system information for a virtual machine using the SCPINFO command and real machine support to physically reconfigure the system using VARY commands. VM/SP also uses the new 3081 hardware instruction TEST BLOCK to validate real storage at system load and initialization. For more information about these topics, see "Section 1. General Considerations."

Elimination of One-Megabyte Segments

Changed: 3081 processors do not permit the use of one-megabyte segments. See "Section 1. General Considerations" for details.

New: The IBM 4341 Model Group 2 offers intermediate system users expanded function and capacity. The 4341-2 features enhanced block multiplexer channel capability and increased data transfer rate. An optional Extended Control Program Support Expansion feature allows concurrent operation of ECPS:MVS and ECPS:VM/370.

3033 Model Group S

New: The IBM 3033 Model Group S processor complex offers large system installations a unique growth advantage within the 3033 processor family. The 3033-S uniprocessor features an increased instruction execution rate for additional processing capability, six channels, and four or eight megabytes of storage.

Miscellaneous

Changed: Various minor technical and editorial changes have been made throughout this publication.

Preface

This publication is for people who run guest operating systems under the VM/SP control program, for example, VSE/AF, VS1/BPE, MVS/SP and VM/SP itself.

Users of the control program (CP) should refer to the *Virtual Machine/System Product CP Command Reference for General Users*, SC19-6211, and *Virtual Machine/System Product Operator's Guide*, SC19-6202.

Users of the conversational monitor system (CMS) should refer to the *Virtual Machine/System Product CMS User's Guide*, SC19-6210.

Users of the remote spooling communications subsystem (RSCS) should refer to the *Virtual Machine Facility/370: Remote Spooling Communications Subsystem (RSCS) User's Guide*, GC20-1816.

Users of the IBM RSCS Networking Program Product (5748-XP1) should refer to the *VM/SP RSCS Networking Program Reference and Operations Manual*, SH24-5005.

Users of the VM/Interactive Problem Control System Extension Program Product (5748-SA1) should refer to the *VM/IPCS Extension User's Guide and Reference*, SC34-2020.

Users of the interactive problem control system (IPCS) should refer to the *IBM Virtual Machine Facility/370: Interactive Problem Control System (IPCS) User's Guide*, GC20-1823.

Note: The recommendations in the DOS and OS areas of this publication are meant to help an installation in generating operating systems to run more efficiently under VM/SP. It contains operational considerations or hints when using virtual machines. Many of these recommendations were suggested by VM/370 users and have not been submitted to any formal test by IBM. As a result, potential users should evaluate the applicability of the recommendations to their installation before implementation.

Generic Terms

Unless otherwise noted the following are the generic terms used throughout the book.

'OS'	is generic for the OS/PCP, OS/MFT, OS/MVT, OS/VS1, VS1/BPE, OS/VS2 SVS, OS/VS2 MVS and MVS/SP operating systems.
'DOS'	is generic for the DOS, DOS/VS, DOS/VS AF, DOS/VSE and VSE/AF operating systems.
'VSE'	is generic for DOS/VSE and VSE/AF.
'OS/VS'	is generic for OS/VS1, VS1/BPE, OS/VS2 SVS, OS/VS2 MVS and MVS/SP.
'POWER'	is generic for the POWER, POWER/VS, VSE/POWER Version 1 and 2 of VSE/POWER Version 2 DOS spooling subsystems.

'VSE/POWER' is generic for Versions 1 and 2 of VSE/POWER.

Organization

This publication contains these sections:

- "Section 1. General Considerations" provides both introductory and general usage information. The introductory information briefly describes the features of VM/SP as they apply both to the virtual machine and to the guest operating system running in it. The general usage information discusses those aspects of running operating systems under VM/SP common to all systems. This section also describes how to use VM/SP functions more efficiently when running guest operating systems under VM/SP.
- "Section 2. VM/SP in a Virtual Machine" explains how to test and update a VM/SP system that is itself operating under VM/SP.
- "Section 3. DOS in a Virtual Machine" provides operating information specific to running DOS in a virtual machine. It contains system planning considerations for using these systems under VM/SP, rather than stand-alone.
- "Section 4. OS in a Virtual Machine" provides operating information specific to running these systems in a virtual machine.

Terminology

In this publication, the following terminology is used:

- "2305" refers to the IBM 2305 Fixed Head Storage, Models 1 and 2.
- "3270" refers to a series of display devices, namely, the IBM 3275, 3276, 3277, 3278, and 3279 Display Stations. A specific device type is used only when a distinction is required between device types.
- "3330" refers to the IBM 3330 Disk Storage, Models 1, 2, and 11; the IBM 3333 Disk Storage and Control, Models 1 and 11; and the 3350 Direct Access Storage operating in 3330/3333 Model 1 or 3330/3333 Model 2 compatibility mode.
- "3340" refers to the IBM 3340 Disk Storage; Models A2, B1 and B2; and, the 3344 Direct Access Storage, Model B2.
- "3350" refers to the IBM 3350 Direct Access Storage, Models A2 and B2, in native mode.
- "FB-512" refers to the IBM 3310 and 3370 Direct Access Storage Devices. These devices are supported by VM/SP.
- "3375" refers to the IBM 3375 Direct Access Storage device.
- "3380" refers to the IBM 3380 Direct Access Storage device.
- "3800" refers to the IBM 3800 Printing Subsystem.
- "Cylinder" describes space on Direct Access Storage Devices (count-key-data-devices) supported by the VM/SP system control program.

- “Block” describes DASD space on FB-512 devices supported by VM/SP.
- The term “area” may appear in the text when there is no need to differentiate between DASD space on count-key-data-devices or FB-512 devices.
- “Records” describes a spool file generated to represent physical card decks.

Any information about display terminal usage also applies to the IBM 3138, 3148, and 3158 Display Consoles when used in display mode, unless otherwise noted.

Any information pertaining to the IBM 3284 or 3286 printer also pertains to the IBM 3287, 3288, and 3289 printers, unless otherwise noted.

Any information pertaining to the IBM 2741 terminal also applies to the IBM 3767 terminal, Model 1, operating as a 2741, unless otherwise specified.

For a glossary of VM/SP terms, refer to *Virtual Machine/System Product Library Guide and Master Index*, GC19-6207.

Prerequisite Publications

Virtual Machine/System Product Introduction, GC19-6200.

Corequisite Publications

You must have a basic knowledge of the operating systems that will be running under VM/SP. For the titles and abstracts of the appropriate publication, refer to the latest *IBM System/370 and 4300 Processors Bibliography*, GC20-0001.

Virtual Machine/System Product

System Programmer's Guide, SC19-6203

Planning Guide and Reference, SC19-6201

CP Command Reference for General Users, SC19-6211

CMS User's Guide, SC19-6210

CMS Command and Macro Reference, SC19-6209

Operator's Guide, GC19-6202

Terminal Reference, GC19-6206

Associated Publications

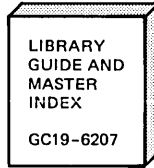
Virtual Machine/System Product OLTSEP and ERROR Recording Guide, SC19-6205

Environmental Recording Editing and Printing (EREP) Program, GC28-1178

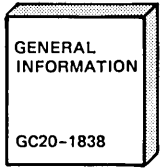
Device Support Facility (DSF) User's Guide and Reference, GC35-0033

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

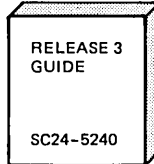
The VM/SP Library



Evaluation



Planning



Installation



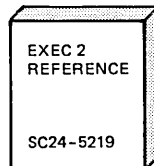
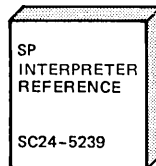
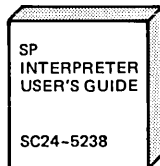
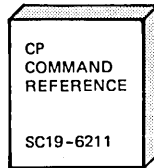
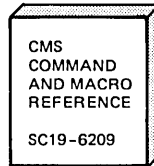
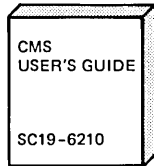
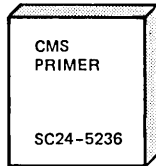
Administration



Operation

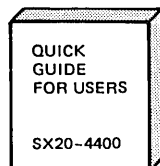


End Use

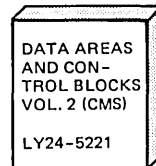
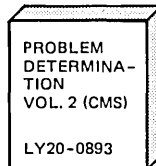
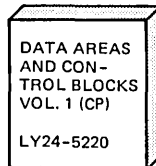
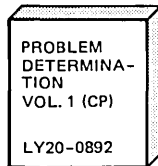
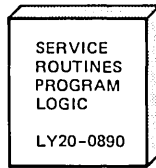


Reference Summaries

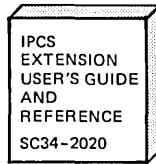
To order all the Reference Summaries, use order number SBOF 3820.



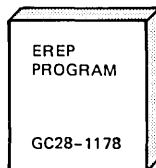
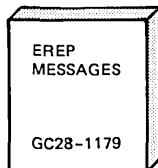
Program Service



Auxiliary Service Support

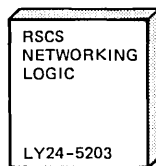
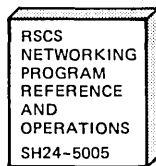
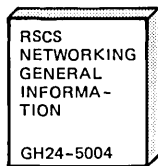


Device Support Facilities
IPCS Extension 5748-SA1

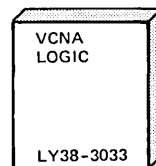
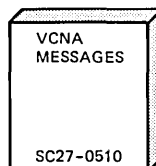
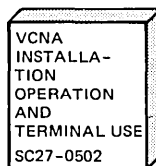
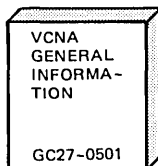


Environmental Recording
Editing and Printing
(EREP)

Auxiliary Communication Support



RSCS Networking
5748-XP1



VTAM Communications
Networking Application
(VCNA) 5735-RC5

Contents

Section 1. General Considerations	1-1
Virtual Machine Resources	1-1
VM/SP Components	1-1
Virtual Machine Operating Systems	1-2
Batch and Single-User Interactive Systems	1-3
Multiple-Access Systems	1-3
Conversational Monitor System	1-3
Other Programs and Systems	1-3
Error Recording and Analysis	1-3
Trace Table Recording Facility	1-4
Unsupported Devices	1-4
Programming Considerations	1-4
Paging Factors	1-5
Reducing Paging Activity	1-5
Abnormal Terminations in a Virtual Machine	1-5
Reducing a Virtual Machine's I/O Operations	1-6
Virtual Machine Options	1-6
ACCT Option	1-6
ECMODE Option	1-6
ISAM Option	1-7
REALTIMER Option	1-8
STFIRST Option	1-8
SVCOFF Option	1-9
Virtual=Real Option	1-9
VMSAVE Option	1-10
370E Option	1-11
BMX Option	1-11
Data Transfer Using VMCF	1-11
Data Transfer Using IUCV	1-11
Set Run On	1-12
BTAM Autopoll Channel Programs	1-12
Controlling Multiple Guest Operating Systems	1-12
Use in a Single System	1-13
Use in Distributed Systems	1-14
Sample Action Routine for Handling VS1 Operator Responses	1-14
Sample Response File for the Action Routine	1-16
Special Considerations for Multiprogramming Systems Under VM/SP	1-16
VM/VS Handshaking	1-16
Handshaking for DOS	1-16
Handshaking for VS1	1-17
The Diagnose Interface	1-17
Page Waits	1-17
I/O Waits	1-17
Spooling	1-18
Spooling Recommendations	1-18
Closing Spool Files	1-19
Processor Model and Channel Model Dependencies	1-19
Dynamic Processor Reconfiguration	1-20
Specifying Virtual Machine Consoles	1-21
Using Specific Devices as Virtual Consoles	1-21
Using a Display Terminal as a Console in Display Mode	1-22
Specifying a Secondary Userid	1-22
Virtual Machine I/O Management	1-23
Dedicated Channels	1-23
Defining Direct Access Storage Devices	1-24
VM/SP Alternate Path Support	1-24
Operating Systems Using DASD Reserve/Release	1-25
Shared DASD	1-25
Virtual Reserve/Release Support	1-26
Restrictions: Device Sharing Between Real Processors	1-26
Restrictions: Device/Minidisk Sharing On a Single Processor	1-26
Example -- Reserve/Release for Dedicated Volumes	1-27
Summary of Reserve/Release Support	1-27
Full Screen Console Support for Virtual Operating Systems	1-28
Switching Modes in a Virtual Machine Environment	1-29
Alternating Between Operating Systems	1-30

Transferring Output	1-30
Configurations for Alternating Between Operating Systems	1-32
Multiple-Access Virtual Machines	1-33
Performance Considerations	1-38
Shadow Table Maintenance Support	1-38
Multiple Shadow Table Support	1-38
Selective Invalidation	1-39
Elimination of One-Megabyte Shadow Tables	1-39
Shadow Table Bypass for V=R Users	1-39
SET STMULTI Command	1-39
Restrictions when Eliminating Shadow Tables	1-39
Shadow Table Bypass for V=V Users	1-40
SET STBYPASS Command	1-40
Restrictions when using Shadow Table Bypass V=V Users	1-42
STFIRST Directory Option	1-42
SET STMULTI Command	1-42
Performance Guidelines	1-45
Configuration Factors Influencing Performance	1-45
Workload Factors Influencing Performance	1-46
VM/SP Performance Factors	1-47
Performance Measurements	1-49
Emphasizing Interactive Response Times	1-49
Generation Procedures Under VM/SP	1-49
Creating VM/SP Directory Entries	1-50
Unique Directory Entry Considerations	1-52
USER Control Statement	1-52
ACCOUNT Control Statement	1-52
OPTION Control Statement	1-52
IPL Control Statement	1-52
CONSOLE Control Statement	1-52
MDISK Control Statement	1-52
SPOOL Control Statement	1-53
DEDICATE Control Statement	1-53
LINK Control Statement	1-54
SPECIAL Control Statement	1-55
Defining Virtual Devices	1-55
AUTOLOG Facility	1-56
Using the CP AUTOLOG Command	1-56
Defining AUTOLOG1 in the Directory	1-56
Multiple Systems With AUTOLOG1	1-57
Sample Directory Entries	1-58
A Multiple-Access Virtual Machine	1-58
Summary	1-59
Section 2. VM/SP in a Virtual Machine	2-1
VM/SP Directory Definition	2-1
Virtual Machine Configuration	2-2
Defining a Console for VM/SP in a Virtual Machine	2-3
CMS System	2-3
Accessing a VM/SP System Running in a Virtual Machine	2-3
CP Disks for the Virtual Machine	2-3
Formatting the Volumes	2-3
Allocating Space for the Volumes	2-4
Virtual IPL and Operation	2-5
Accessing Devices	2-5
Graphic and Spool Devices	2-6
Virtual Disks	2-6
Spooling Considerations	2-6
Example -- VM/SP Under VM/SP	2-7
Summary	2-20
Section 3. DOS in a Virtual Machine	3-1
System Generation Recommendations	3-1
VM/SP Recommendations	3-2
DOS Recommendations	3-2
DOS Accounting	3-4
Sample DOS Directory Entries	3-5
Accessing DOS	3-6
Using Virtual Unit Record Devices	3-6

Defining the Operator's Console	3-7
Preparing Jobs for a DOS Virtual Machine	3-7
Loading DOS	3-8
IPL Using ASI	3-8
IPLing Using a Saved System	3-13
IPL from the Console	3-13
IPL from the Card Reader	3-17
DOS Operation	3-18
Starting a Job Stream (W/O POWER)	3-18
When a Job Is Finished	3-19
Communicating with CP	3-19
Interrupting the Virtual Machine	3-20
Running Batch DOS Under VM/SP	3-20
Alternating Between CMS and DOS Under VM/SP	3-21
Loading CMS into a Virtual Machine	3-22
Using the System Product Editor	3-22
Issuing SPOOL Commands To Control Unit Record Devices	3-23
Punching CMS Files	3-23
Initializing DOS	3-23
Signaling DOS To Read the Job Stream	3-24
When the Card Reader Is Empty	3-24
Reloading CMS into a Virtual Machine	3-25
Examining DOS Virtual Machine Output	3-25
Using EXEC Procedures	3-25
Using More Than One Virtual Machine	3-27
Accessing CMS	3-27
Disconnecting CMS	3-28
Logging Onto DOS	3-28
Returning to CMS	3-29
Disconnection Considerations	3-30
Sending Jobs to a Disconnected DOS Machine	3-30
Console READs and WRITEs in a Disconnected DOS Machine	3-31
Developing and Testing Programs to Run in a DOS Virtual Machine	3-31
Summary	3-33
Section 4. OS/VS in a Virtual Machine	4-1
System Generation Recommendations	4-1
VM/SP Recommendations	4-2
IPL Command Enhancement	4-2
OS/VS Recommendations	4-3
OS Recommendations	4-3
VM/VS Handshaking for VS1	4-4
Activating VM/VS Handshaking for VS1	4-5
VS1 Nonpaging Mode	4-5
Closing CP Spool Files	4-7
VS1 Without Handshaking	4-8
BTAM Autopoll CCW Change Detection	4-8
Miscellaneous Enhancements	4-8
IBM 3850 Mass Storage System Considerations	4-9
Sample OS/VS Directory Entries	4-9
Accessing OS/VS	4-10
Using Virtual Devices	4-11
Defining the Operator's Console	4-12
Using the VM/SP Spool File System	4-12
Preparing Jobs for an OS/VS Virtual Machine	4-12
Job Entry and Output Retrieval	4-13
Loading OS/VS	4-13
OS/VS Operation	4-14
Communicating with CP	4-16
Using the #CP Function	4-16
Using OS/VS in Batch Mode Under VM/SP	4-16
Alternating Between CMS and OS/VS Under VM/SP	4-17
Loading CMS into a Virtual Machine	4-17
Using the CMS Editor To Prepare Job Streams	4-18
Issuing Spool Commands to Control Unit Record Devices	4-18
Punching CMS Files	4-18
Initializing OS/VS	4-19
Reloading CMS into a Virtual Machine	4-19
Examining OS/VS Virtual Machine Output	4-19

Dynamic SCP Transition to or from Native Mode	4-19
Restriction for MVS Virtual Machines	4-20
Operating Procedures	4-20
Error Recording	4-22
Using More Than One Virtual Machine	4-22
Disconnection Considerations	4-23
Sending Jobs to a Disconnected OS/VS Machine	4-23
Console READs and WRITEs in a Disconnected OS/VS Machine	4-23
Developing and Testing Programs to Run in an OS/VS Virtual Machine	4-24
OS Uniprocessor Under VM/SP	4-25
Use of Single Processor Mode in AP and MP Systems	4-25
Operating Procedures	4-25
Restrictions	4-27
Error Recording	4-28
Taking a VM/SP Dump	4-28
Summary	4-29
Index	X-1

Figures

1-1. Virtual=Real Machine	1-10
1-2. Summary of VM/SP Reserve/Release Support	1-28
1-3. OS Job Stream Transfer	1-31
1-4. Directory Entry for Alternating Between Operating Systems	1-32
1-5. Virtual Devices: Local 3270 Terminals	1-33
1-6. Virtual Devices: Remote Terminals	1-34
1-7. A Virtual VM/SP Multiple-Access System	1-35
1-8. Directory Entry for a Multiple-Access Virtual Machine Running VM/SP under VM/SP ..	1-35
1-9. A Communications Test System	1-36
1-10. A Virtual 2703 TCU Controlling Remote 3270 Terminals	1-37
1-11. Relationship of VM/SP Directory Entries	1-51
1-12. AUTOLOG1 Virtual Machine and PROFILE EXEC	1-57
1-13. Sample Directory Entry for Multiple-Access TSO System	1-58
2-1. VM under VM: Sample Directory Entry for TESTSYS	2-1
2-2. VM under VM: Logon procedure	2-7
2-3. VM under VM: Verifying the Virtual Machine Configuration	2-7
2-4. VM under VM: Accessing a virtual machine's minidisk	2-8
2-5. VM under VM: Redefining the Console	2-8
2-6. Linking 333 from read only to write	2-17
3-1. IPLing DOS Using Automatic System Initialization (ASI)	3-10
3-2. IPLing DOS Using the Saved System Facility of VM/SP	3-13
3-3. IPL DOS and Execute a Job in the Card Reader	3-16
4-1. Sample IPL of VS1 under VM/SP	4-15

Section 1. General Considerations

The Virtual Machine/System Product (VM/SP) provides an easy, convenient way to use a single terminal to run guest operating systems, such as DOS, OS, or VM/SP. With VM/SP, users can test a new application program with an operating system, or they can develop and test new operating system releases. This work can be done isolated from any work that is running concurrently elsewhere in the system. This isolation is achieved by the use of a virtual machine.

A virtual machine is a functional equivalent of an IBM System/370 computing system. Each virtual machine has the functional equivalent of a real processor, main and auxiliary storage, and I/O devices. Because VM/SP only simulates these functions, this simulated machine is referred to as a “virtual” machine. VM/SP manages the functions of a real IBM System/370 in such a way that virtual machines are available to multiple concurrent users.

Before running any operating system in a virtual machine, an installation should consider:

- How application programs can operate efficiently in a virtual machine.
- How it can reduce a virtual machine’s I/O operations.
- Which services are available for performance and communication for both VM/SP and a virtual machine.
- What special considerations there are for multiprogramming operating systems under VM/SP, such as DOS or OS.
- What operating system functions and devices VM/SP supports and does not support.
- How virtual machines can access the VM/SP system.

This publication assumes you have a basic understanding of VM/SP concepts and functions as described in the *VM/SP Introduction*. It is also assumed that you have a basic understanding of whatever operating system you are running under VM/SP.

Virtual Machine Resources

Virtual machine resources can be shared or alternated between users for specific time periods. To have resources allocated to any particular virtual machine, they must be specified in the VM/SP directory entry for that virtual machine. The directory entries of all the virtual machines makes up the VM/SP directory file. The directory file is usually located on the VM/SP system residence volume. When a user obtains access to the VM/SP system, a virtual machine is created based upon that user’s directory entry. The user can then load any of the supported operating systems and begin processing.

VM/SP Components

VM/SP has two components:

1. The Control Program (CP) controls the resources of the real processor to provide multiple virtual machines.

2. The Conversational Monitor System (CMS) provides a wide range of conversational and time-sharing facilities. By using CMS you can:
 - a. Create, update and manage files
 - b. Compile, test and execute problem programs

When you install VM/SP in conjunction with the VM/SP Release 6 System Control Program, it becomes a functional operating system that provides extended features to the System Control Program and CMS System components of VM/370 Release 6.

You can appreciably expand the capabilities of VM/SP by installing the RSCS Networking program product (5748-XP1) and the VM/IPCS Extension program product (5748-SA1).

For an overview of these VM/SP concepts, refer to *VM/SP Introduction*.

Virtual Machine Operating Systems

While the control program (CP) of VM/SP manages the concurrent execution of virtual machines, it is also necessary to have an operating system managing the work flow within each virtual machine. Each virtual machine executes independently of every other virtual machine, and they can each use:

- The same operating system
- A different operating system
- Different releases of the same operating system

The operating systems that can execute in virtual machines are:

Batch or Single-User Interactive

DOS
 DOS/VS
 DOS/VS AF
 DOS/VSE
 VSE/AF
 OS/PCP
 OS-ASP
 OS/MFT
 OS/MVT
 OS/VS1
 VS1/BPE
 OS/VS2 SVS
 OS/VS2 MVS
 MVS/SP
 RSCS

Multiple-Access

VM/SP
 VM/Systems Extensions
 VM/Basic System Extensions
 VM/370
 Time Sharing Option of OS VSE with VSE/ICCF (5746-1 TS1)

Conversational

CMS

CP provides each of these operating systems with virtual device support and virtual storage. The operating systems themselves execute as though they are controlling real devices and real storage, but they must not violate any of the restrictions listed in the *VM/SP Planning Guide and Reference*.

Batch and Single-User Interactive Systems

When operating in a virtual machine under VM/SP, the user has the choice of running either 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. This may decrease the overall efficiency of the virtual machine. When running single partitions in multiple virtual machines, the need for multiple virtual storage spaces places a burden on auxiliary storage. However, using shared systems (when possible) reduces this burden.

Multiple-Access Systems

Each multiple-access system operates in one virtual machine and supports multiple interactive users. The multiple-access virtual machine must first gain access to VM/SP (by using the LOGON command). Subsequently, interactive users can connect to the multiple-access system (by using the DIAL command or by using a terminal on a dedicated line). Communication between the two is carried out by using the command language of the multiple-access system.

Conversational Monitor System

The Conversational Monitor System (CMS) is a VM/SP component that provides a wide range of conversational and time-sharing facilities. Together with the control program of VM/SP, it provides a time-sharing system suitable for direct problem solving and program development. By using CMS a virtual machine user can:

- Create, update and manage files
- Compile, test and execute problem programs

The CMS interactive capabilities are extended to DOS environment users by using either the CMS/DOS environment or CMS. For OS users, a combination of CMS commands and CMS simulation of OS access methods and SVCs provides similar interactive capabilities.

For information on using the CMS virtual machine, refer to the *VM/SP CMS User's Guide* and the *VM/SP CMS Command and Macro Reference*.

Other Programs and Systems

For information about other programs and systems that have been used under VM/SP, request information on Installed User Programs (IUP's), Program Products (PPs), and Field Developed Programs (FDPs) from your local IBM branch office. For a list of these programs, refer to the *VM/SP Planning Guide and Reference*.

Error Recording and Analysis

The operating systems that are commonly run in virtual machines all use SVC 76 to write error records to the error recording data sets. However, in a virtual machine VM/SP intercepts SVC 76 and records the error in its own error recording area. Therefore, error records from all operating systems reside in this one centralized

error recording area. To access the recorded data, use the CMS CPEREP command. For further information about error recording, formatting output from the error recording area with Service Record File devices, and CPEREP refer to the *VM/SP OLTSEP and Error Recording Guide*.

Trace Table Recording Facility

Problem determination capability is expanded to service personnel and system programmers with the Trace Table Recording Facility. The facility uses the CP command CPTRAP to create a chronological record of selected trace table, virtual machine interface, and CP interface information on a READER spool file. Use of the facility is intended for the analysis of VM/SP problems that escape detection using a system dump.

A CMS utility program (TRAPRED) is included as part of the CPTRAP facility. TRAPRED uses the READER file as input, and supports output to either a spooled print file or an interactive terminal display. For additional information on using the CPTRAP facility, refer to the *VM/SP System Programmer's Guide*.

Unsupported Devices

Virtual machine users may be able to use I/O devices that VM/SP does not support. An unsupported device is a device type that is not listed under the DEVTYPE operand of the RDEVICE macro instruction in the *VM/SP Planning Guide and Reference*. To use an unsupported device, a user must attach or dedicate the device to a virtual machine. A dedicated device is one that is used exclusively by one user. However, VM/SP supports these dedicated devices only under these conditions:

- No timing dependencies exist in the device or the program.
- No dynamically modified channel programs exist in the access method, except when OS ISAM or OS TCAM Level 5 are used.
- No special functions need to be provided by VM/SP.
- None of the other CP restrictions are violated. (Refer to the VM/SP restrictions listed in the *VM/SP Planning Guide and Reference*.)
- The device is generated into the VM/SP nucleus (by using the RDEVICE macro instruction with the appropriate CLASS operand).

I/O devices that are part of a virtual machine's configuration require real device equivalents. However, the exceptions to this rule are:

- Unit record devices that VM/SP can simulate by using spooling techniques.
- Virtual 2311 disks that VM/SP maps onto 2314 or 2319 disks. One to two full 2311 units can be mapped onto a 2314 or 2319 disk in this manner.

Programming Considerations

New application programs should be designed to operate efficiently in a paging environment. Whenever possible, use VM/SP paging instead of DOS or OS paging. That is, make the DOS partitions and OS regions virtual=real (V=R) and large enough to contain the largest jobs. Eliminate all overlays, and if possible, combine into one large job any multistep jobs that use temporary DASD storage.

Paging Factors

Installations should be aware that the following factors affect the performance of a virtual machine:

- The frequency of real interruptions that occur
- The frequency and type of privileged instructions executed
- Whether the virtual machine assist or VM/370 extended control-program support hardware is on the machine and enabled both the system operator and the user
- The frequency of START I/O (SIO) instructions
- Locality of reference for paging activity within virtual storage
- The amount of fixed head paging space
- The location of the paging areas on DASD
- Whether the enhanced page migration scheme polls preferred paging areas using moveable heads

These factors are in addition to those described under the topic “Performance Guidelines” discussed in this chapter.

Reducing Paging Activity

When a virtual machine refers to virtual storage addresses that are not in real storage, a page fault (and paging activity) occurs. Routines that have widely scattered storage references tend to increase the paging load caused by this virtual machine.

When possible, modules dependent upon each other, as well as the related reference tables, constants, and literals, should be located in the same 4K page. Infrequently used routines, such as those that handle unusual error conditions, should not be placed near main routines. To minimize paging, reentrant coding techniques should be used whenever possible.

Abnormal Terminations in a Virtual Machine

In VM/SP there are three levels of storage:

- First level storage - real storage
- Second level storage - virtual storage that VM/SP creates and manages for a virtual machine
- Third level storage - virtual storage that a virtual storage operating system (such as MVS) creates and manages when running in a virtual machine

Whenever possible use the virtual storage operating system’s dumping procedure instead of VM/SP’s. The CP dump program does not print out third level storage pages (that is, V=V regions or partitions of OS and DOS machines) in the correct sequence. Pages that happen to be stored on the OS or DOS paging disk are not printed at all. Refer to *VM/SP CP Command Reference for General Users* for

information about CP dump commands. Also, several special formatting dump programs are available to help you trace through DOS and OS control blocks. For more debugging information, refer to the *VM/SP System Programmer's Guide*.

Reducing a Virtual Machine's I/O Operations

The number of start I/O instructions (SIO) executed by a virtual machine may be substantially reduced by:

- Using 4K byte blocking factors for I/O areas
- Preallocating the DASD space for OS or OS/VS work data sets
- Using virtual storage instead of DASD work files for smaller temporary files
- Building temporary files in virtual storage and letting VM/SP page out the data (if needed)
- Omitting virtual printers, punches, and readers from each partition or region in a virtual machine because records for these devices are unblocked
- Using the virtual operating system's spooling subsystem (such as POWER/VS or JES) because these spooling subsystems use large I/O areas and long chains of CCWs

Virtual Machine Options

VM/SP provides several optional services to virtual machines. Specify these options either in the OPTION control statement of the VM/SP directory program, or in the CP SET command.

The REALTIMER, ISAM, 370E and ECMODE options increase the amount of VM/SP overhead incurred by the virtual machines using them. Therefore, do not specify them for a virtual machine unless they are required. If a particular situation requires an option only occasionally, use the CP SET command and not the OPTION statement. In this way, the additional overhead is incurred only while the option is in effect. For more information about specifying these and the other options in the OPTION control statement, refer to the topic "Creating VM/SP Directory Entries" discussed in this chapter.

ACCT Option

The ACCT option allows one user to charge another user for virtual machine resources. For example, if the machine is performing a batch type operation, a virtual machine user can generate job accounting records for each job processed. To generate accounting records for a virtual user, refer to the *VM/SP System Programmer's Guide* for a discussion of DIAGNOSE code X'4C'.

ECMODE Option

The ECMODE option allows the virtual machine to use the complete set of virtual System/370 control registers and the dynamic address translation feature of the System/370. Programming simulation and hardware features are combined to allow use of all the available features in the hardware. This option is required for all virtual storage operating systems. It is also required when executing the generalized trace facility (GTF) under OS.

Specify the ECMODE option if the virtual machine uses an operating system that:

- Runs in extended control mode
- Uses dynamic address translation (DAT)
- Uses extended control registers other than zero
- Addresses I/O channels 6 through 15

If this option is not specified in the directory, a user can enter EC mode by issuing the CP SET command with the ECMODE operand:

```
#cp set ecmode on
```

When the ECMODE option is specified for a virtual machine, the saved segments of the virtual operating system can be shared.

Note: Setting the ECMODE option does not alter the ECMODE bit of the user's PSW.

ISAM Option

The ISAM option allows the virtual machine to execute the self-modifying CCW command sequences generated by the OS ISAM modules in OS, MFT, or MVT. This option is not required for the proper functioning of ISAM in DOS or OS. However, the ISAM option is required under one of these two conditions:

- ISAM is run in the virtual=real area of an OS virtual machine.

----- or -----

- VM/VS handshaking is active.

The ISAM option does not permit other types of self-modifying CCW sequences to function.

Certain ISAM channel programs that execute under OS, or in a V=R region of OS/VS use a self-modifying operation not allowed under normal VM/SP processing. With the ISAM option selected, VM/SP can scan the specific ISAM channel program to handle the self-modifying sequence properly.

Only those users with the ISAM option in their VM/SP directory entry, or who have issued the CP SET ISAM ON command, have their CCW strings checked for self-modifying operation; thus, not all users incur the additional VM/SP overhead. This option is not needed for DOS and OS ISAM when run in a V=V region.

The ISAM option must be specified if a user is:

- Using ISAM in an OS virtual machine
- Using ISAM in a V=R partition or region of an OS/VS virtual machine
- Using VS1 handshaking with VS1 nonpaging

Do not specify the ISAM option if a user is:

- Using ISAM in a DOS virtual machine
- Using ISAM in a V=V region of an OS/VS virtual machine

REALTIMER Option

This option is required for operating systems running applications (such as CICS) where certain interruptions are timer driven. Normally a virtual interval timer indicates only the real CPU time used by the virtual machine (virtual CPU time). The REALTIMER option updates the virtual machine interval timer when that virtual machine is in a self-imposed wait state. This way both real CPU time and wait time used are indicated. As a result of the way in which the virtual interval timer at location X'50' functions, it does not provide accurate time of day whether or not the REALTIMER option is specified. OS PCP, MFT, and MVT operating systems for System 360 and DOS Version 3 for System/360 and System/370 use the interval timer at location X'50' for time of day values, and therefore, will not obtain accurate times.

Enter the REALTIMER option if the virtual timer is to be updated during virtual wait time as well as during virtual processor time. If this option is not specified in the directory entry, a user can obtain this timing facility by issuing the CP SET command with the TIMER operand:

```
#cp set timer real
```

To turn off the option, issue:

```
#cp set timer off
```

STFIRST Option

When virtual machine assist is available on the machine, the STFIRST option permits the virtual machine user to issue the SET STBYPASS *nnnnnK* command to initiate shadow table support.

The SET STBYPASS *nnnnnK* command defines the high-water mark (or highest limit) of the virtual operating system's nucleus that is mapped guest virtual=guest real. That is, the area of the virtual machine's storage that will not be used by the guest operating system for paging. This specification allows VM/SP to reduce the overhead associated with maintaining shadow page and segment tables under CP. Thus, VM/SP can now selectively invalidate shadow table entries associated with pages stolen from within this guest virtual=guest real area, rather than invalidating the entire table.

The STFIRST option should be used for virtual machines that execute production workloads on these virtual operating systems: DOS, OS, SVS, and MVS. Only specify STFIRST for virtual machines executing test systems or programs that follow the programming restrictions for shadow table bypass. (These restrictions are listed under the topic "Shadow Table Maintenance Support" in this chapter.)

Warning: Otherwise, either VM/SP could abnormally terminate, or third to first level storage mapping could be disrupted.

SVCOFF Option

The SVCOFF option specifies that VM/SP, rather than virtual machine assist or ECPS:VM/370, handles all SVC interruptions. To override this option, issue the CP SET command:

```
#cp set assist svc
```

Note: If the operating system uses SVC 76 for error recording, VM/SP handles the SVC 76 interruptions whether SVCOFF is in effect or not.

Virtual=Real Option

The virtual=real option may be desirable, not possible, or mandatory depending on the following conditions:

- It is desirable when running a virtual machine operating system (like OS/MVS) that performs its own paging because it eliminates the possibility of double paging.
- The virtual=real option cannot be used when running an operating system such as DOS or VS1 in nonpaging mode with VM/VS handshaking in a virtual machine.
- It is mandatory to use the virtual=real option to allow programs that execute self-modifying channel programs or have hardware timing dependencies to run under VM/SP.

The virtual=real area is set up at VM/SP IPL time. The primary VM/SP system operator can release the area for use as part of the dynamic paging area. Once released, it cannot be reclaimed except by shutting down and reIPLing VM/SP. The virtual=real area must be released in total; that is, unused pages of the area cannot be selected for release.

There are several ways to use the virtual=real option effectively on a data communication system with no CCW translation (SET NOTRANS ON) that has multiple ports. Dedicate either the transmission control unit or communications lines to that system via the ATTACH command or by the VM/SP directory DEDICATE statement. Conversely, on a multiple port data communication virtual=real operation, virtual 270x lines (that is, lines assigned and used by the CP DEFINE and DIAL commands) operate with CCW translation. When VM/SP detects the use of nondedicated communication lines, it ignores the SET NOTRANS ON command. (See Figure 1-1 on page 1-10.)

Note: By issuing the SET STBYPASS VR command, a user eliminates shadow tables and the overhead associated with maintaining them. VM/SP modifies the virtual operating system's page table to relocate virtual page zero to the first 4K page following the V=R area. It is then possible to dispatch the virtual machine and have VM/SP's real control register 1 point to the virtual machine's page and segment tables. To terminate the shadow table bypass function, issue the SET STBYPASS OFF command. For details about how to specify this command, refer to the *VM/SP CP Command Reference for General Users*.

A virtual machine operating system (OS or VM/SP) running in the V=R area of a 3081 processor can improve its reliability and availability by using the TEST BLOCK instruction (TB) to validate its storage. For more information about the

TEST BLOCK instruction, specifying a virtual=real machine, and defining the VIRT=REAL option in a virtual machine's directory entry, refer to the *VM/SP Planning Guide and Reference*.

By specifying the virtual=real option, the virtual machine is eligible to occupy VM/SP's low storage. With the exception of page 0, all other virtual storage addresses correspond to the real storage addresses. VM/SP's page 0 occupies the first 4096 bytes of real storage, and VM/SP moves virtual page 0 to a position immediately following the area set aside for V=R operation (see Figure 1-1).

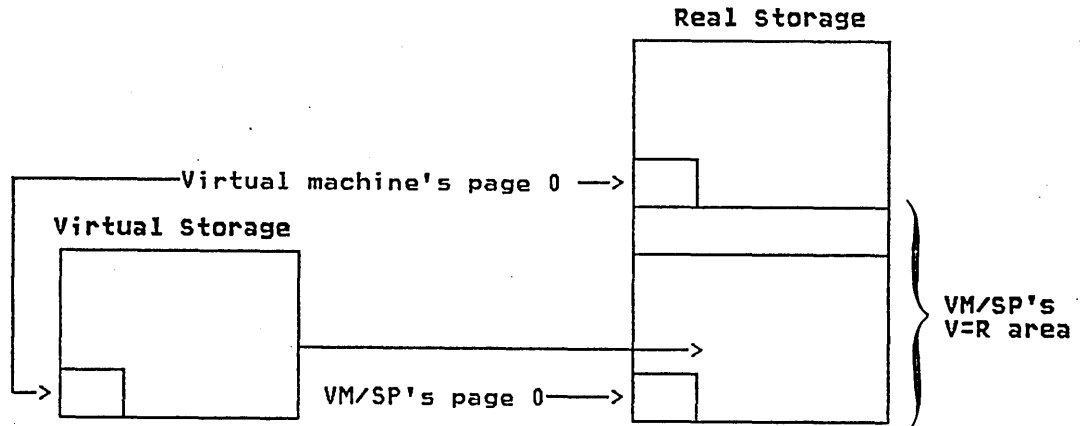


Figure 1-1. Virtual=Real Machine

This option can be specified for many virtual machines; however, only one virtual machine can occupy the V=R area at any one time. If you specify V=R when the V=R area is already occupied, VM/SP creates the virtual machine in virtual=virtual mode and informs you of this development.

VMSAVE Option

If a virtual machine user specifies the VMSAVE option (in conjunction with predefined values in the NAMESYS system generation macro instruction), VM/SP automatically saves the virtual machine's contents. This is done either when VM/SP terminates the virtual machine or when VM/SP itself abnormally terminates. After logon, a user can restore the contents by issuing the IPL command. In this command, the user specifies the name of the save area that preserves the contents of the virtual machine. While the VMSAVE option can be used by any virtual machine, it is most useful for production workloads on data base/data communication systems such as IMS.

To specify the VMSAVE option, either use the VMSAVE directory option or issue the SET VMSAVE *area-name* command. For details about how to use the VMSAVE option, refer to the *VM/SP System Programmer's Guide*. For details about how to use the SET VMSAVE *area-name* command, refer to the *VM/SP CP Command Reference for General Users*.

370E Option

The 370E option lets a user specify whether a virtual machine can use the MVS/System Extensions or MVS/System Product functions of the VM/SP Program Product¹. Enable the MVS/System Extensions or MVS/System Product functions by:

- Specifying the 370E option in the OPTION control statement of the virtual machine's directory.

----- or -----

- Issuing the class G command SET 370E ON, provided the system operator has issued the class A command SET S370E ON.

To disable these functions for the VM/SP system, system operators can issue the class A command SET S370E OFF. To disable these functions for a specific virtual machine, general users can issue the class G command SET 370E OFF.

To display the ON and OFF system status of the MVS/System Extensions or MVS/System Product functions, both classes of users can issue the QUERY S370E command. To display the ON and OFF status of these functions for a specific virtual machine, general users can issue the class G command QUERY SET.

Note: Some processors allow the coexistence of virtual machine assist and the S/370 Extended Facility (S370E) and S/370 Extended Feature (See your local IBM Sales Office). Thus, an MVS/SE or OS virtual machine can run under VM/SP with virtual machine assist active on a 158-3 processor.

BMX Option

The BMX (virtual block multiplexer) option allows an operating system running in a virtual machine to overlap multiple SIO requests on a specified channel path. The selector channel mode is the normal (and default) channel mode for virtual machines. When the BMX option is given control, it applies to all channels in the virtual machine, except to channel 0. This option can be specified regardless of whether block multiplexer channels are attached to the processor. The CP DEFINE command can redefine the channel mode for a virtual machine.

Data Transfer Using VMCF

Virtual machines can communicate and exchange data with other virtual machines by using the virtual machine communication facility (VMCF). VMCF is the interface among communicating virtual machines. To initiate a VMCF function, the operating system in the virtual machine must issue a DIAGNOSE instruction. For a detailed description of the DIAGNOSE instruction, refer to the *VM/SP System Programmer's Guide*.

Data Transfer Using IUCV

Virtual machines can communicate and exchange data with other virtual machines by using the inter-user communications vehicle (IUCV). This is accomplished

¹ These program products support the System/370 Extended Facility and the System/370 Extended Feature. For a list of the appropriate processors supported by the Facility and the Feature, refer to the *VM/SP Planning Guide and Reference*.

through the use of the IUCV macro instruction. IUCV allows messages to be presented to the virtual machine by polling functions or via external interrupts. For a detailed description of both IUCV functions and the IUCV macro instruction, refer to the *VM/SP System Programmer's Guide*.

Set Run On

Unless you are using the debug facilities of CP, your guest operating system's virtual machine should have issued the CP SET RUN ON command.

Note: Some operating systems with VM/VS handshaking, (such as VSE/AF), issue the CP SET RUN ON command at initialization time via the DIAGNOSE X'08' CP interface.

BTAM Autopoll Channel Programs

If an operating system is executing BTAM channel programs, VM/SP checks each BTAM autopoll CCW string to see if it has been dynamically changed. This is done every time the string is executed. To bypass this checking, issue the CP command:

```
set autopoll on
```

Whenever the BTAM autopoll CCWs are modified, OS Release 6 and DOS Release 34 with the Advanced Functions-DOS Program Product (Program No. 5746-XE2) use the DIAGNOSE instruction code X'28' to notify VM/SP.

The combination of the SET AUTOPOLL ON command and the use of the diagnose interface reduces VM/SP overhead and improves the overall performance for that particular user. However, both of these facilities must be active.

Warning: If a user has specified SET AUTOPOLL ON and the operating system does not use the diagnose interface, a channel program modification goes undetected.

The results are unpredictable.

If a user has specified SET AUTOPOLL OFF and the operating system uses the diagnose interface, this unnecessary checking results in performance degradation.

Controlling Multiple Guest Operating Systems

Users may have found that running more than one production guest operating system requires an operator for each system. With the help of the Programmable Operator Facility, an installation can now use one operator to control all of the guest systems. The programmable operator facility reduces the workload of the CP system operator in a stand-alone VM/SP system, and limits the need for a trained VM operator in a distributed VM/SP system. The programmable operator facility can be programmed to automatically execute a predetermined command or set of commands for the operator. It can route messages intended for the system operator to a person designated as the **logical operator**. The logical operator is the person whose virtual machine receives all of the messages the programmable operator facility is not programmed to handle.

The programmable operator facility permits an operator located at a terminal (locally attached or remotely attached to a host system) to handle most operations in the same manner as if the CP system operator were seated at the system console. Manual operations such as flipping the console switches or tape mounting must still

be done manually, but the logical operator need not be skilled in other aspects of computer operations. In order to help facilitate distributed processing, the programmable operator facility is generally run on the CP system operator console; however, it can be run on any virtual machine.

Assume an installation has three VS1 virtual machines. These VS1 virtual machines are running disconnected and their console output is being sent to another virtual machine (called OSOPER) via the Single Console Image Facility. The programmable operator facility is executing in OSOPER; therefore, PROP intercepts and handles every message being written to any one of the VS1 virtual machines. Each message received by the programmable operator facility is put into a CMS disk file called the **log file**. The messages are identified by the date and time received, and every day a separate log file is started. Some of the messages are purely informational while others are important for tracking purposes.

The programmable operator facility compares all messages directed to it against entries listed in the **routing table**. The routing table is a separate CMS file that specifies the action to take for each message received and authorizes certain users to invoke specific PROP commands. When a match between the message sent and an entry in the routing table occurs control is passed to the specified **action routine** (see the example on the next page). If a message has no entry in the routing table then it is sent to the logical operator. The logical operator takes action necessary to respond to the message. The necessary actions might be to mount a tape or disk, or to instruct the programmable operator facility to issue a SEND command to the VS1 virtual machine that originated the message.

With the appropriate routing table the messages going to the logical operator can be held to a minimum. The number of guest operating systems controlled by a single logical operator depends on the number of messages handled by the programmable operator facility.

The programmable operator facility is designed for:

- use in a Single System
- use in Distributed Systems.

| Use in a Single System

When the programmable operator facility is operational in a single-system environment it can:

- Ease message traffic to the system operator by:
 - Filtering non-essential (information only) messages
 - Routing messages to the logical operator for specific actions
- Increase productivity by freeing the system operator of routine responses

Only essential messages are sent to the logical operator for response or action.

Use in Distributed Systems

Once installed, the programmable operator facility can:

- Make responses and perform tasks that do not require an on-site operator
- Filter non-essential (information only) messages
- Route messages requiring on-site (manual) intervention to the logical operator
- Allow the logical operator at the host site to:
 - control the Programmable Operator Facility operation
 - control the distributed system
- Allow one operator to control a network of systems.

Sample Action Routine for Handling VS1 Operator Responses

A set of action routines are provided with the programmable operator facility; but, other action routines may be written to perform specific functions unique to each installation. New programmable operator commands may be added simply by the addition of a new action routine. Following is an example of an action routine written to handle VS1 operator responses and invokes the response file OSCMD LIST.

```

/*****
*
* This Exec is to be used in conjunction with the Single Console Image
* Facility (SCIF) and a guest operating system (GOS). The user who
* 'owns' this Exec is the secondary user of the GOS. All messages
* appearing on the 'console' of the GOS virtual machine, are reflected
* to the secondary user running the programmable operator. The
* programmable operator facility processes all incoming message text and
* parameters found in the PROP RTABLE. If the incoming message text and
* RTABLE parameters match an entry in the response list, that entry will
* be executed by the GOS. Otherwise, the message will be passed to the
* logical operator for handling.
*
*****/

    traceo; address cms;

/*****
*
* Assign meaningful variable names to the arguments passed to the 'Exec'.
* Also, get the RTABLE parameters and the message text from the console
* stack passed to the 'Exec'.
*
*****/

    arg requester reqnode lopr loprnode msgtype prop proprnode netmach rtable x;
    pull msgtext;
    pull rtableparms;

/*****
*
*                               Process the incoming message.
*
*****/

    'STATE OSCMD LIST';                               /* Does response list exist? */
    if rc = 0 then do                                  /* If so, find a matching message. */
        ;EXECIO * DISKR OSCMD LIST (FINIS LOCATE /' || word(msgtext,1);

    if rc = 0 then do                                  /* If a match, then get correct */
        pull linenum;                                  /* response from console stack and */
        pull msg response;                             /* execute response in the primary */
        'CP SEND' requester response;                 /* virtual machine. */
    end;

/*****
*
* If no match is found, pass the message to the logical operator.
*
*****/

    else 'EXEC TELL' lopr 'AT' loprnode msgtext;

/*****
*
* If there is no response list, pass the message to the logical operator.
*
*****/

    end;
    else 'EXEC TELL' lopr 'AT' loprnode msgtext;
    exit;                                             /* Return control to the */
                                                    /* programmable operator facility */

```


Sample Response File for the Action Routine

This is the response file OSCMD LIST searched by the above action routine. It contains the responses necessary to initialize a VS1 and VCNA machine. The identifiers of the VS1 requests appear in the left column and the responses to the messages appear to the right.

```
*IEA000A      R 0,HN
IEF868I       S VTAM.PO,,, (LIST=2B)
IST020I       S VCNA.P1
*01           V NET,ACT,ID=luid,LOGON=applid
```

For detailed information on the use of the programmable operator facility see the *VM/SP System Programmer's Guide*.

Special Considerations for Multiprogramming Systems Under VM/SP

When a multiprogramming operating system such as OS or DOS is run in a virtual machine, its resource-management algorithms interact with those of VM/SP -- especially when the virtual operating system has a page wait or an I/O wait. Multiprogramming operating systems use these two methods to interact with VM/SP:

- VM/VS handshaking
- Diagnose interface

This topic discusses the above methods and the unique situations that apply when running an operating system under VM/SP.

VM/VS Handshaking

VM/VS handshaking permits instructions issued by an operating system in a virtual machine to be processed directly by the processor. It also permits VM/SP to simulate privileged instructions. VM/VS handshaking is available for these operating systems running in virtual machines under VM/SP:

- DOS/VS Release 34 with the Advanced Functions DOS/VS Program Product (5746-XE2)
- VSE with the VSE/Advanced Functions Program Product (5746-XE8)
- VS1 Release 4 and subsequent releases

Handshaking for DOS

DOS Release 34 with the Advanced Functions-DOS Program Product (5746-XE2) uses handshaking. For further details, refer to the appropriate DOS program product publications. VSE with the VSE/Advanced Functions Program Product (5746-XE8) uses VM/VS handshaking (also known as the VSE-VM/370 linkage facility). For further details, refer to *The VSE System General Information*.

Handshaking for VS1

Although handshaking is a system generation feature for VS1, it is active only when VS1 is under the control of VM/SP. It is disabled when that same VS1 operating system is run on a real machine. For details about VM/VS handshaking for VS1, refer to the "OS in a Virtual Machine" section in this publication.

The Diagnose Interface

The diagnose interface (the DIAGNOSE instruction) permits operating systems running in virtual machines under VM/SP to communicate easily and efficiently with VM/SP.

By inserting DIAGNOSE instructions where appropriate in the operating system's code, several functions can be requested by a virtual machine. Details on how to use the DIAGNOSE instruction to request these functions are in the *VM/SP System Programmer's Guide*.

Page Waits

If during its execution, an OS or DOS created task or program must wait for a VM/SP service such as a virtual storage page, VM/SP marks the virtual machine nondispatchable. This occurs even if other partitions or tasks in that virtual machine may be ready and available for processing. Those other tasks in the virtual machine cannot be dispatched by the operating system until the VM/SP page wait is satisfied. Thus, the highest priority program of the virtual machine gets almost all of the processor time allocated to that virtual machine, if it can use the time. Therefore, programs running in the other partitions experience *significant degradation*.

When multiprogramming systems must be executed in a virtual machine, make the partitions or regions as large as practical and execute all jobs V=R. Also, consider using the VM/SP virtual=real option, reserved page frames, or locked pages. When using the VM/SP virtual=real option, it eliminates paging for one virtual machine, but this may adversely affect the paging performance of other virtual machines. The reserved page frames option tends to keep the most active pages in storage, and the locked pages option locks the specified pages in storage.

Remember: If the region size is made too large, certain programs (such as the OS sort/merge program) do not run efficiently.

I/O Waits

On a real machine, when a task is waiting for an I/O operation to complete, the lower priority tasks are given use of the processor. Under VM/SP, the I/O operations of a particular virtual machine are overlapped with the processor execution of that and other virtual machines. Consequently, lower priority tasks created by OS and DOS are given the processor resource less frequently when executing in a virtual machine than when executing in a real machine.

To set the priority of a virtual machine, the VM/SP system operator can issue the CP command SET PRIORITY. A low priority value gives the virtual machine a higher priority, and this priority ensures that VM/SP dispatches the virtual machine for execution more frequently than other virtual machines.

To ensure that the lower priority DOS or OS tasks have a chance to execute, installations can use the favored execution option. This option reduces the effect of a variable system load on the favored virtual machine. It allows an installation to modify the normal VM/SP scheduling algorithms and forces VM/SP to devote more of its processor time to a given virtual machine. The option causes VM/SP to keep the specified virtual machine in the active queue, unless it becomes nonexecutable. To obtain this option for a specific virtual machine, the system operator must issue the CP SET FAVORED command.

To guarantee that a certain amount of processor time is made available to a virtual machine, installations can use the favored execution option with the percentage value specified in the SET FAVORED command. For information on the use of the SET QDROP command, refer to the *Operator's Guide* and for more details about the performance options refer to the *VM/SP System Programmer's Guide*.

Spooling

Most multiprogramming operating systems, such as DOS and OS, have their own spooling subsystems (such as VSE/POWER). Because VM/SP also provides its own spooling, double spooling can occur. This raises the following questions. Should an installation:

- Use only the operating system's spooling subsystem?
- Use only VM/SP's spooling?
- Use double spooling?

If an installation has a significant amount of printing or punching to do, it may appear that one of the spooling subsystems should be eliminated. This is not necessarily true. In fact, if the multiprogramming operating system's spooling subsystem blocks its output (as does VS1), the most efficient spooling arrangement is usually to let both VM/SP and VS1 spool.

Note: Virtual operating systems cannot take advantage of the RAS enhancements contained in 3800 hardware engineering change level 454846 (unless the 3800 is attached to a guest machine). VM/SP ignores this engineering change level at real print time on a 3800.

Spooling Recommendations

If DASD space is not a limiting factor, use double spooling. If possible, generate a stripped-down version of the virtual machine's spooling subsystem, eliminating those functions not used by that virtual machine. Make the I/O buffer sizes as large as possible to cut down on SIO instructions.

If an installation has only enough DASD spooling space for one spooling subsystem and if only one virtual machine generates significant amounts of spooled output, then let that virtual machine do the spooling. However, if many virtual machines spool data and must use a common pool of unit record devices, then an installation should probably let VM/SP do the spooling.

Closing Spool Files

Output spool files are not scheduled for the real printer or punch devices until one of these actions occur:

- The user logs off, or VM/SP forces the user off.
- The user issues the CP SYSTEM CLEAR command.
- The user loads an operating system via the CP IPL command.
- The user (either manually or through an EXEC procedure) issues the CP CLOSE command to close the spool file.
- VS1 or VSE/POWER with handshaking uses the diagnose interface to issue the CP CLOSE command after each job completes.
- The installation modifies the operating system by adding DIAGNOSE instructions to communicate with VM/SP to close the spool files.

One of the above actions must occur before the spool files are placed in the CP unit record queues. To keep the spool files from building up on the spooling DASD, close the spool files periodically. For instance, close them at the end of each job.

Processor Model and Channel Model Dependencies

Channel checks (such as channel data checks, channel control checks, and interface control checks) no longer cause the virtual machine to be reset. Thus, an operating system that now runs in a virtual machine can attempt to recover from a channel check or to terminate in an orderly manner.

If channel error recovery procedures in an operating system depend on the processor model and channel model, then these two requirements must be met:

1. Depending upon the recovery procedures of the specific operating system running in the virtual machine, an installation may have to generate the operating system for the same processor model on which VM/SP is to run.
2. The virtual machine configuration must have each virtual channel correspond to a single type and model of real channel.

The second requirement means that all virtual devices on a virtual channel must correspond to real devices on real channels. The real channels must be identical to each other in type and model. For example, assume that a virtual machine has a 3330 disk at virtual address 280 and a 3340 disk at virtual address 290 that correspond to similar real devices on real addresses 380 and 590, respectively. Because both virtual devices (280 and 290) are on a single virtual channel (channel 2), the corresponding real devices (380 and 590) must both be on real channels that have an identical channel type and model. By meeting this requirement, when an operating system issues a STIDC (store channel ID) instruction to virtual channel 2, VM/SP can simulate it the same way and returns consistent results to the operating system.

Not only should the real channels be identical, but generally speaking, they should be of the same type as the virtual channel. (The virtual channel type is defined

either in the OPTION statement in the virtual machine's directory entry or by the class G CP DEFINE CHANNEL command.) Two exceptions to this general rule are:

- When the real channel is a block multiplexer channel, the virtual channel can be a selector channel. In this case, the simulated STIDC instruction returns this information to the operating system: (1) the model number of the block multiplexer channel, and (2) a channel type field indicating that the channel is operating in selector mode.
- When the real channels are selector channels, the virtual channel can be a block multiplexer channel. This specification may improve performance when the virtual channel has devices on several real selector channels. It allows the virtual machine to overlap channel operations on the virtual channel and to take full advantage of the several selector channels. However, when VM/SP simulates the STIDC instruction issued to the virtual block multiplexer channel, it gives the operating system the channel type and model number of a selector channel, not of a block multiplexer channel. While this result is inconsistent with the channel's operation as a block multiplexer, the operating system should not detect or be affected by this inconsistency. For further restrictions about channel model-dependent functions, refer to the *VM/SP Planning Guide and Reference*. processor model dependencies model dependencies

Dynamic Processor Reconfiguration

In special operation environments, such as Attach Processor and Multiprocessor Environment (AP/MP), certain dynamic processor configuration exists.

The 3082 Processor Controller is a support processor that monitors and supervises on-going operational activity and performs a central communication function for the 3081 processor complex. The monitoring and service support facility (MSSF) is a hardware component of the processor controller. MSSF provides system configuration and storage information for the 3081 processor complex. The MSSF CALL diagnose instruction allows VM/SP to communicate with the MSSF. MSSF CALL is a diagnose instruction with a function code of X'0080'.

Virtual operating systems that are able to communicate with the MSSF can use the SCPINFO command to query processor configuration and storage allocation. If the SCPINFO command is issued on an MVS virtual machine operating in V=V mode, CP simulates the MSSF response and returns predefined response codes to the virtual machine. If the MVS virtual machine is operating in V=R mode, the MSSF hardware processes the request and returns the information to the virtual machine unaltered.

VARY PROCESSOR commands that modify the real processor configuration are also processed by the MSSF to physically bring the processor online or offline. When a VARY ONLINE PROCESSOR or VARY OFFLINE PROCESSOR VPHY command is issued by the VM/SP operator on a 3081 processor, an MSSF CALL instruction is generated to the MSSF. The MSSF services the request and a completion status code is returned. If you are using single processor mode on a 3081 processor, use the VARY OFFLINE PROCESSOR VLOG command to logically vary the processor offline. If the FORCE or VPHY option of the VARY command is used, the processor will be physically (electronically) varied offline and is unavailable to the MVS operating system when MVS is IPLed in the V=R area.

Specifying Virtual Machine Consoles

To specify more than one console for a virtual machine, the virtual machine user must tell VM/SP about the existence of these additional consoles. Operating systems may support either:

- *Multiple consoles* -- where different classes of system messages can be routed to different consoles.

----- or -----

- *Alternate consoles* -- where the user can switch to a backup console if the primary console becomes inoperative.

To tell VM/SP about the existence of these consoles, either use directory statements or issue CP commands. The way a user specifies the second console depends upon whether:

- The user always wants to use a specific device at a specific I/O address.

----- or -----

- The user wants flexibility in selecting which device or terminal is to be used.

Using Specific Devices as Virtual Consoles

Assumption:An OS/VS virtual machine is to run its 3158 display console at address 01F in display operator console mode. Also, the virtual machine is to operate a local 3270 terminal at address 1B8.

Step 1: Generate OS/VS with at least two consoles. Use 01F as the primary console, and 009 as the secondary console.

Step 2: Specify the secondary console by using the VM/SP CONSOLE directory statement. Code it:

```
CONSOLE 009 3210
```

Step 3: Specify the OS/VS primary console either by having a DEDICATE statement in the VM/SP directory or by using the CP ATTACH command after logon. Either specification allows OS/VS to use the 3158 console in display operator console mode.

If the DEDICATE statement is used, then code it:

```
DEDICATE 01F 01F
```

If the ATTACH command is used -- then -- after logon, send a message to the VM/SP operator that requests the following ATTACH command to be issued:

```
msg op attach 01f to userid as 01f
```

Using a Display Terminal as a Console in Display Mode

To use a 3270 display terminal as the OS or DOS primary console in display operator console mode, either have a **SPECIAL** statement in the virtual machine's VM/SP directory entry or issue the **CP DEFINE GRAF** command after logon to VM/SP.

If the **SPECIAL** statement is used, it appears as follows:

```
SPECIAL 01F 3270
```

If the **SPECIAL** statement is not used, assume that a local 3270 line has been enabled by the VM/SP operator. Then, issue the following **DEFINE** command:

```
define graf 01f 3270
```

In either situation, after you log onto VM/SP (by using the device specified in the **CONSOLE** statement) and load the operating system into the virtual machine (by using the **IPL** command with the **STOP** option), you must issue the **CP DIAL** command at the 3270 console that is to be used in display mode. This action logically connects that 3270 console to the operating system.

Notes:

1. As an alternative to the above method, the **CP TERMINAL CONMODE 3270** command may be used to obtain a display mode console for your guest operating system. The **CP TERMINAL CONMODE 3270** command is not supported for 3270 terminals going through a VTAM Service Machine.
2. The **DIAL** function is not supported for terminals controlled by a VTAM service machine.

If the second console is a remote terminal such as a 2741 or 3767 connected by either a 2702 or a 3704/3705 in 2702 emulation mode, the **SPECIAL** statement would appear as follows:

```
SPECIAL 01F 2702 IBM
```

The **DEFINE** command would be:

```
define line as 01f ibm
```

Specifying a Secondary Userid

Normally, a disconnected user has no console services. However, by specifying a secondary userid on the **CONSOLE** directory control statement, a virtual machine can retain console services while disconnected. Any console output created by the disconnected virtual machine or CP on behalf of the disconnected virtual machine transfers to the console of the secondary user. Also, using the **CP SEND** command, the secondary user can issue commands and replies on behalf of the disconnected user. For further information on the **SEND** command, refer to the *VM/SP CP Command Reference for General Users*.

Note: In order to use this secondary user support, the virtual console of the disconnected user must be used as a 3215 console. Full-screen-display type operations are rejected, causing a "command reject" situation. Therefore, the disconnecting virtual machine must be out of full screen mode before issuing the **CP DISCONNECT** command.

Virtual Machine I/O Management

A real disk device and a real or emulated 270x transmission control unit (TCU) can be shared among multiple virtual machines. Virtual disk device sharing is specified in the VM/SP directory entry or by a user command. A specific virtual machine may be assigned read-only or read/write access to a shared disk device. To gain access to the shared virtual device, a user must supply the appropriate password. To ensure device integrity, VM/SP checks each virtual machine I/O operation against the parameters in the virtual machine configuration.

The virtual machine operating system is responsible for the operation of all virtual devices associated with it. These virtual devices may be defined in the VM/SP directory entry of the virtual machine, or they may be attached dynamically to (or detached from) the virtual machine's configuration for the duration of the terminal session.

Virtual devices may be in one of these three states or conditions:

- *Dedicated* -- when mapped to a fully equivalent real device
- *Shared* -- when mapped to a minidisk or when specified as a shared virtual disk device
- *Spooled* -- when VM/SP places the device output on intermediate direct access storage

For example, in a real machine when running under control of the operating system, the problem program requests the system to issue a SIO instruction to a specific device. The operating system normally initiates the I/O operation and handles any device error recovery. In a virtual machine, the operating system performs the same functions, but the device address specified and the storage locations referenced are both virtual. VM/SP has the responsibility for translating the virtual specifications to real.

When I/O interruptions occur, VM/SP reflects them to the virtual machine for its interpretation and processing. When I/O errors occur, VM/SP records the error but it does not initiate error recovery operations. The operating system must handle error recovery.

When VM/SP initiates I/O operations for its own paging and spooling, the operation is not subject to translation, and VM/SP itself performs the operation.

Dedicated Channels

In most cases, many virtual machines share the I/O devices and control units on a channel both as minidisks and dedicated devices and with VM/SP system functions such as paging and spooling. Because of this sharing, VM/SP has to schedule all the I/O requests to achieve a balance among virtual machines. In addition, VM/SP must reflect the results of the subsequent I/O interruption to the appropriate storage areas of each virtual machine.

By specifying a dedicated channel for a virtual machine (using the class B ATTACH CHANNEL command), the virtual machine has the channel and all the devices for its own exclusive use. For dedicated channels, VM/SP translates the virtual storage locations specified in channel commands to real locations and

performs any necessary paging operations. However, VM/SP does not translate a device address because the virtual device addresses on the dedicated channel must match the real device addresses; thus, minidisks cannot be used.

Dedicated devices should be considered as an alternative to dedicated channels because then the only translation done by VM/SP is device address translation. Dedicated devices should be all put on the same channels so that VM/SP can handle them more efficiently for a virtual machine.

Defining Direct Access Storage Devices

Virtual machines can use DASD as either minidisks or dedicated volumes. A real disk volume can be shared by several virtual machine users, each owning a number of adjoining cylinders or blocks. This logical subdividing of a real disk volume is called physical pack sharing, and each subdivision of cylinders or blocks is called a virtual disk or minidisk. A real disk volume can also be dedicated to a specific virtual machine for its own private use. When using dedicated disk volumes, the virtual machine's operating system must perform all necessary interruption handling, error recovery, and error recording.

By using either the LINK directory control statement or the CP LINK command, a user can share the data on a minidisk or entire disk volume with the owner of the virtual disk. The LINK statement or command allows controlled, concurrent access to the data on the virtual disk. This sharing is called logical data sharing.

If any virtual machine temporarily requires additional direct access space, the user can use the CP DEFINE command to obtain it dynamically from a pool of temporary (T-disk) space. To define a pool of T-disk space, an installation specifies the size of the T-disk pool when allocating disk space with the stand-alone CP Format/Allocate program.

Before using the T-disk space, the user must first initialize it. For storing DOS, OS, or VSAM files, use the Device Support Facility program, (provided as part of VM/SP), to initialize the minidisk and set up the VTOC. Refer to the *Device Support Facilities*. For CMS files, issue the CMS FORMAT command. For CP disks, use the Format/Allocate utility (DMKFMT) to initialize disks to be used by a guest VM/SP system. Temporary minidisks are available to the virtual machine for the duration of the current terminal session. The area is returned to VM/SP when one of the following actions occur:

- The system forces the virtual machine off.
- The virtual machine logs off.
- The user issues a CP DETACH command to release the temporary minidisk.

For details about defining, formatting, using, and sharing minidisks, refer to the *VM/SP Planning Guide and Reference*.

VM/SP Alternate Path Support

VM/SP alternate path support permits the definition of alternate paths to a tape or DASD unit on the VM/SP processor; this option supports the two/four channel switch and string switch features. Define alternate paths in VM/SP's DMKRIO for devices that a virtual operating system is to use. When you do, VM/SP will map I/O requests from a virtual address associated with the virtual machine to one of the real paths to the device as defined in DMKRIO. Refer to the section, "Alternate Path Support" in the *VM/SP Planning Guide and Reference* for an explanation on the definition of alternate paths.

As a rule, alternate path and reserve/release support are mutually exclusive. There is one exception to this rule. At the minidisk level, VM/SP provides virtual reserve/release support in the form of a software locking mechanism. Alternate paths can be defined for the real device as long as virtual machines under VM/SP use virtual reserve/release between themselves and the real processors do not share the volume. Refer to the section, "Operating Systems Using DASD Reserve/Release" for specific information on DASD sharing and virtual reserve/release.

For specific information on channel switching, refer to the *VM/SP Planning Guide and Reference*.

Operating Systems Using DASD Reserve/Release

Reserve/release CCW commands prevent several users of the same data files from simultaneously accessing the same data. This is most useful when the data is being updated. While VM/SP handles the reserve/release CCW commands presented by other operating systems running in virtual machines, VM/SP and CMS do not use reserve/release CCW commands. Operating systems use these commands under two conditions:

1. When running in virtual machines under VM/SP and sharing data files
2. When running on other processors and sharing data files with operating systems that run under VM/SP

VM/SP has two types of reserve/release support:

- **Shared DASD** -- applies to virtual machine operating systems that issue reserve/release CCWs to preserve data integrity; the hardware reserve preserves the data integrity on a device basis.
- **Virtual Reserve/Release** -- applies to virtual machines issuing reserve/release CCW commands to minidisks that are designated as subject to virtual reserve/release processing.

Shared DASD

Reserve/release support allows several operating systems (such as MVS, SVS, and VS1) to have data protection on a full volume, regardless if they are running as virtual machines under VM/SP or on other processors. The hardware reserves the device when a reserve CCW command is executed. VM/SP supports reserve/release CCW commands for shared DASD as though each virtual machine has a separate channel path to a shared device.

Note: When a reserve is issued to a device that has alternate path support (defined in the RDEVICE and RCTLUNIT VM/SP system generation macro instructions), VM/SP changes a reserve CCW command to a sense CCW command.

Virtual Reserve/Release Support

Virtual reserve/release support allows several operating systems (such as MVS, SVS, and VS1) to do the following:

1. To run all virtual machines under the same VM/SP operating system
2. To have data protection when using the same data files on the same minidisk.

To use virtual reserve/release, specify 'V' in the mode operand of the MDISK directory statement. Also subject to virtual reserve/release processing are the virtual machine users who use the same minidisk by way of LINK statements.

By using the VM/SP virtual reserve/release support, one operating system running in a virtual machine can prevent other operating systems running under the same VM/SP system from accessing the reserved minidisk. However, a minidisk protected by virtual reserve/release support may not be protected from access by an operating system running on other processors.

Restrictions: Device Sharing Between Real Processors

- When a device is shared between processors and at least one of the processors is running VM/SP, the shared volume cannot contain more than one minidisk. The single minidisk may encompass the entire volume or a small portion of the volume. Neither CP nor any virtual machine may reference the remainder of the volume for use as paging, spooling, etc. device.
- Devices shared between processors must not be generated in VM/SP's DMKRIO as having alternate paths. It may happen that there are multiple paths from the VM/SP processor to the shared devices and a path from these shared devices to another processor. If so, you may not use the ALTCH or ALTCU macro instruction operands to define alternate paths from the VM/SP processor. This means that the definition of alternate paths in DMKRIO and the use of real reserve/release are mutually exclusive.

Restrictions: Device/Minidisk Sharing On a Single Processor

- If more than a single path to a volume exists, DMKRIO may be generated so that each path is defined as a separate path, not as an alternate path. When this is done, each path can be attached or dedicated to a different user, and reserve/release CCWs issued by such users preserves the data integrity. In this case, the integrity is preserved by the hardware, not by the software reserve/release support. Again the definition of alternate paths in DMKRIO and the use of real reserve/release are mutually exclusive.
- A volume may be defined through the directory to contain one or more minidisks. Such minidisks must be identified through the MDISK statement as requesting virtual reserve/release support. These minidisks may then be shared between virtual machines that support Shared DASD (not CMS) and the data integrity will be preserved by the use of reserve/release CCWs in the virtual machine channel program. Alternate paths may be defined to the device when using virtual reserve/release. The reserve CCW will still be changed to a sense CCW but the integrity will be preserved by the virtual reserve/release code.

Example -- Reserve/Release for Dedicated Volumes

In this example, 230 and 330 are alternate device addresses for a particular DASD to be shared by USERA and USERB (two virtual machines running on the same real computing system that supports sharing). To share this device:

1. Generate the operating systems for virtual machines USERA and USERB to support DASD sharing (for example, RESERVE/RELEASE), and the DASD at addresses 230 and 330 respectively.
2. Generate VM/SP as though 230 and 330 were different devices (with different control units and channels).
3. Issue the CP ATTACH command to attach device 230 to USERA and device 330 to USERB.

Note: If the system generated for USERB is to run in a real machine, rather than a virtual machine, you must:

1. Generate the VM/SP system with device 230 but not 330
2. Issue the CP ATTACH command to attach device 230 to USERA

In both cases:

- The device addresses generated for systems to run in a virtual machine *need not be the same* as on the real machine.
- The devices used by virtual machines *must be dedicated* (attached or defined with a DEDICATE statement in the VM/SP directory).

Do not share the CP SYSRES and any other CP-owned disk between two processors.

Summary of Reserve/Release Support

VM/SP checks all CCW commands passed by operating systems running in virtual machines. It bases reserve/release CCW command processing on:

- The type of device
- The presence or lack of alternate path support
- Whether the MDISK statement in the VM/SP directory contains a “V” in the mode operand

Depending upon the various combinations of these items, VM/SP either permits the reserve CCW command to execute on the hardware or changes the reserve CCW command to a sense CCW command. To determine the conditions when a “reserve” is changed to a “sense” CCW command, refer to Figure 1-2 on page 1-28.

Type of Device	Alternate Path Support	Reserve/Release Executes in the Hardware	Virtual Reserve/Release Requested (V Added to Mode in MDISK)	CCW Comnd Sent by VM/SP to Device	Note
Dedicated DASD or Tape	Not defined	Not applicable	Not applicable	Reserve	1
	Defined	Not applicable	Not applicable	Sense	1
Minidisks	Not defined	Yes	No	Reserve	1
	Not defined	Yes	Yes	Reserve	1
	Not defined	No	No	Reserve	3
	Not defined	No	Yes	Sense	4
	Defined	Not applicable	Not applicable	Sense	2

- ¹ Normal Operation -- The command is passed unchanged to the hardware.
- ² When the VM/SP system has been generated with alternate path support for those devices, it prevents the devices from being reserved. This action causes VM/SP to avoid a possible channel lockout. VM/SP does not return any indication of this action to the operating system issuing the CCW command that the device was not reserved.
- ³ Without the reserve/release capability, VM/SP sends the release/release CCW command unchanged to the hardware. However, the hardware rejects the command and does not reserve the device.
- ⁴ Before sending the command to the hardware, VM/SP changes the reserve CCW command to a sense CCW command and places a virtual reserve on the minidisk. The real device is not reserved. The virtual reserve prevents other operating systems running under the same VM/SP system from accessing the minidisk. However, these same virtual operating systems may reserve other minidisks located on the same real volume. Because the two-channel switch feature is not installed on the channels, only one address path goes to the device from the VM/SP processor. This path allows VM/SP virtual reserve/release processing to send a sense CCW to the device, although the reserve CCW command would be rejected by the hardware.

Figure 1-2. Summary of VM/SP Reserve/Release Support

Full Screen Console Support for Virtual Operating Systems

VM/SP provides full screen console support for virtual machine operating systems that can take advantage of the full screen mode of operation. This support removes the need to attach an additional 3270 terminal. A user can logon a local 3270 display and alternate between full screen mode (graphic device) and CP mode (line device).

A virtual console operator determines when to switch between CP mode and full screen mode. In addition, the virtual console operator has the option of saving the full screen image before switching to CP mode. Upon returning to full screen

mode, the saved screen image is restored. For a further description of mode switching, refer to the CP TERMINAL command in the *VM/SP CP Command Reference for General Users*.

Switching Modes in a Virtual Machine Environment

If the virtual operating system needs a printer, the virtual console operator can perform the following steps:

1. Enter CP mode (via the PA1 key).
2. Ask the operator to dedicate a printer to the virtual machine.

```
msg op please attach the 3800 printer as 001
```

3. Wait for the operator to answer your request.

```
sleep
```

The response to the operator message is:

```
PRINTER 001 ATTACHED
```

4. Return to full screen mode and type:

```
begin
```

5. Initialize and start the printer so that the virtual machine operating system can take advantage of it.

With this facility in VM/SP, an installation does not have to sacrifice two terminals in order to run an operating system that relies on full screen support. For a detailed description of full screen console support, see the *VM/SP System Programmer's Guide*.

Notes:

1. Although 3270 terminals with different screen sizes are supported, the virtual operating system is responsible for issuing the correct 3270 start I/O commands.
2. Although the virtual operating system can use a local logon 3270 terminal for full screen support, the guest cannot get an attention interrupt from the break-in key since CP uses it for mode switching.
3. Spooled console files contain no information entered while in full screen (3270) mode. However, this problem can be solved by console spooling under control of the guest operating system. This avoids the loss of any information.
4. In order to use CMS, the console must be in line (3215) mode. In fact, issuing the IPL CMS command automatically resets the console to line mode.

Alternating Between Operating Systems

A virtual machine user may require the facilities of more than one operating system during a single terminal session.

When running an operating system from a terminal, use the System Product Editor to create and modify job streams and to analyze the results and output.

Application programmers who normally use CMS to interactively create, modify, and test programs, may require facilities for compilation or execution that are not supported or available in CMS.

The technique described in this topic uses multiple operating systems consecutively. Job control cards, compiler or assembler source programs, and test data streams are created and modified at the terminal under control of the System Product Editor. The job stream is then executed, by passing control to an appropriate operating system that has the necessary facilities.

In this way, the programmer uses the terminal-oriented facilities of CMS to create and update source programs and JCL. When ready to compile or test, the programmer can give control of his virtual machine to the operating system. After execution is finished, he can give control back to CMS to selectively scan and display printer and punch output at the terminal.

This approach assumes that the programmer has created source program files and data files under CMS. To execute under another operating system (in this example OS), the programmer must also create JCL records that specify the compilation, link editing, or execution, as appropriate. These records are created under CMS and named with a distinctive filename and filetype (for example, PLICOMP JCL). Job control records, source program files, and data files can then be merged together in the virtual card reader to form a single OS job stream. The CP and CMS commands (shown in Figure 1-3 on page 1-31) create and transfer this job stream.

Transferring Output

The CP SPOOL command transfers subsequent (not currently existing) card images from the virtual card punch of one virtual machine to the virtual card reader of that same or some other virtual machine. During this time, no real cards are punched or read; VM/SP manages the transfer of CMS card-image data files through disk spooling operations only.

Figure 1-3 on page 1-31 shows how to punch files to the virtual machine's card reader. The virtual machine is in the CMS environment at the start of the example. To assist you in distinguishing the commands from the system responses, **the executable commands are highlighted.**

```

CMS
cp close 00c
cp purge 00c all
cp spool 00d nocont purge
cp spool 00d to * cont
punch jobcard jcl (noheader)
punch plicomp jcl (noheader)
punch plimain pli (noheader)
punch asmcomp jcl (noheader)
punch asmsub assemble (noheader)
punch linkgo jcl (noheader)
punch godata dat (noheader)
punch slshstar jcl (noheader)
cp spool 00d nocont close
cp spool 00c cont eof
cp ipl 230

```

Note: The following are issued once under OS control.

```

IEE007A READY
set date=xx.355,Q=(231)
start rdr,00c
start wtr,00e
start

```

Figure 1-3. OS Job Stream Transfer

WHERE:

The command 'cp spool 00c cont eof' specifies that reading is continuous until all files spooled to the virtual machine are exhausted, and the virtual end-of-file button on the reader is pushed.

'noheader' specifies that no special control cards are to be inserted at the beginning of each punched file.

Virtual device 230 is an OS system volume.

Virtual device 231 contains the OS job queue, SYS1.SYSJOBQE.

Note: In Figure 1-3, all standard CMS and OS responses have been omitted. Only the OS READY message is included to illustrate the IPL sequence. Assuming that the user has a 2741, the attention key must be pressed before entering each OS command. The attention interruptions are not shown in the figure.

To transfer files between systems, the user must have access to both operating systems being used. Access to both systems can be provided either in the virtual machine's VM/SP directory entry, or dynamically before loading the new system.

Figure 1-4 on page 1-32 illustrates a virtual machine configuration and the corresponding VM/SP directory control statements. Virtual device addresses 190


```

USER OS2 PASSWORD 1M 2M G
ACCOUNT NUMBER BIN16
OPTION REALTIMER ECMODE BMX
CONSOLE 01F 3215
SPOOL C 2540 READER
SPOOL D 2540 PUNCH
SPOOL E 1403
LINK JFK 230 230 R
LINK MAINT 19E 19E RR
LINK MAINT 19D 19D RR
LINK MAINT 190 190 RR
MDISK 191 3330 101 10 UDISK1 WR RPASS WPASS
MDISK 231 3330 097 043 OSUDSK WR RPASS WPASS

```

Figure 1-4. Directory Entry for Alternating Between Operating Systems

Configurations for Alternating Between Operating Systems

Users can alternate between operating systems easily if the following two conditions are met:

1. Devices used by both systems are supported at the same device address.
2. Common addresses are not used to support different devices.

If the preceding two conditions are not met, the user must modify the virtual machine configuration before each IPL of a new system.

If the two systems require online typewriter keyboards at different addresses, use the CP DEFINE command to change the address of the virtual system console.

If the systems expect different device types at the same address, the common address must be assigned to the appropriate device each time a new system is loaded. If CMS is running with a disk at address 191 and OS is generated to support a 3330 at that address, issue the following command before loading OS:

```
cp detach 191
```

An appropriate device can then be added to the virtual machine at address 191. Add the device either before loading or in response to a mount request from the OS system.

Notes:

1. For direct access storage devices, this procedure is necessary even if both systems support the same device type at the same address. Except for VSAM disks, the disk format used by CMS is unique. It is not compatible with that of other operating systems. Files can be passed between CMS and OS or DOS only through VM/SP spooling or through VSAM data sets.
2. CMS can read DOS and OS sequential data sets, and OS Partitioned Data Sets (PDSs).

Multiple-Access Virtual Machines

Multiple-access programs execute in a virtual machine and directly control terminals. These terminals do not have to be supported by VM/SP as virtual operator consoles, but they may be of any type supported by the program executing. These programs use lines that are either dedicated to the virtual machine (by the directory entry) or assigned to the virtual machine dynamically.

Figure 1-5 shows two multiple-access systems (controlled by virtual machines VM1 and VM2). While each system controls real 3277s by using part of the real 3272, the real 3272 appears to both virtual machines as though they each have sole control of it. (The virtual system consoles of VM1 and VM2 are not shown.)

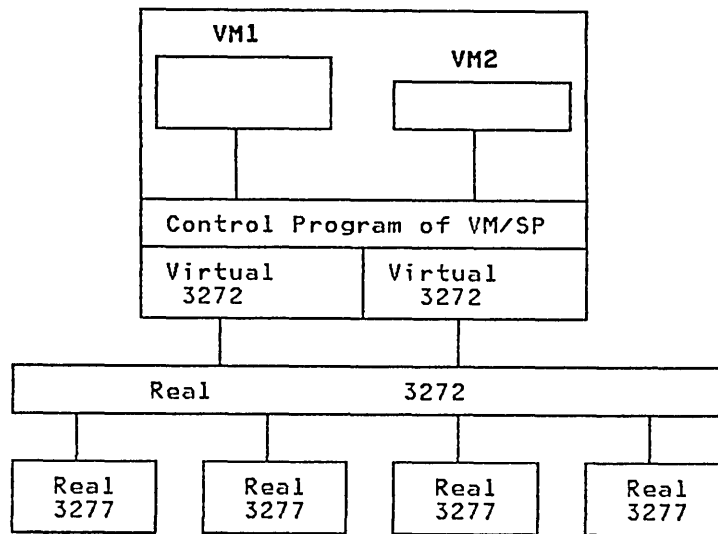


Figure 1-5. Virtual Devices: Local 3270 Terminals

Note: Users can define virtual lines for a virtual machine. These lines are a subset of the lines controlled by a real transmission control unit (TCU).

A subset of the lines of a real transmission control unit (TCU) can be defined as virtual lines for a virtual machine, as shown in Figure 1-6 on page 1-34 .

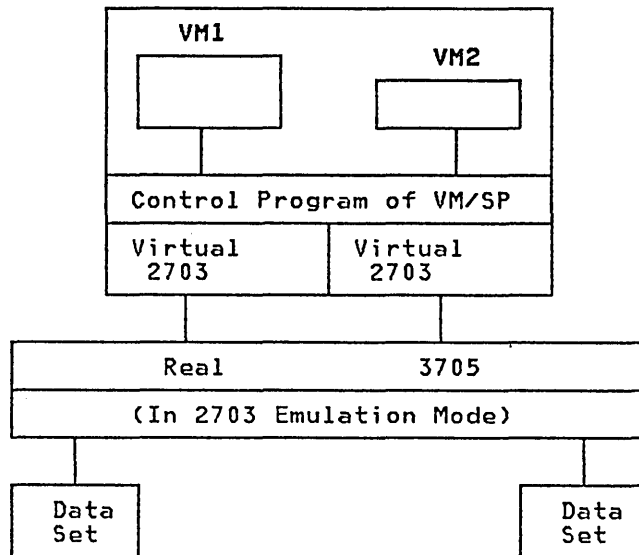


Figure 1-6. Virtual Devices: Remote Terminals

Note: Two lines on the real 3705 are defined as virtual lines for two virtual machines named VM1 and VM2. The remaining lines may support virtual operator consoles.

As shown in Figure 1-7 on page 1-35, virtual machine operating systems may be one like VM/SP itself, or TSO, that supports a number of remote terminals.

To assign a real line as a virtual line, the terminals supported by the virtual machine's operating system are of the same type as those supported by VM/SP as virtual system consoles. To make this assignment, define the virtual lines either in the virtual machine's VM/SP directory entry (via the SPECIAL control statement) or add them to the logged-on virtual machine (via the CP DEFINE command).

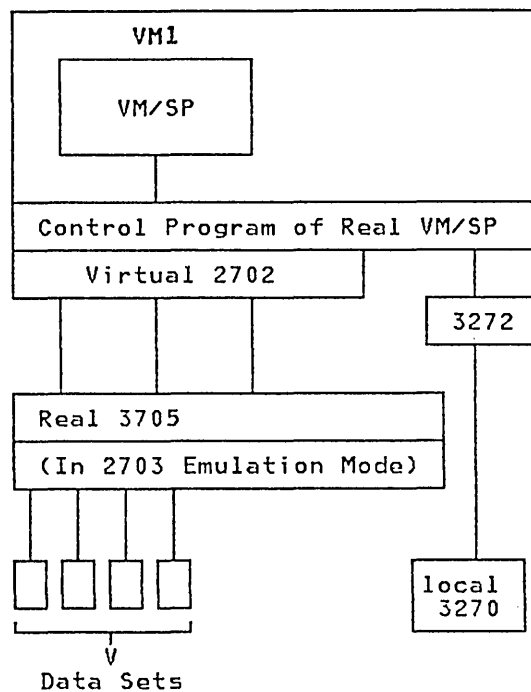


Figure 1-7. A Virtual VM/SP Multiple-Access System

Figure 1-8 illustrates a directory entry for a multiple-access virtual machine to run VM/SP under VM/SP.

```

USER TESTVM PASSWORD 1M 16M G
OPTION REALTIMER ECMODE BMX
IPL 150
CONSOLE 01F 3215
SPOOL 00C 2540 READER
SPOOL 00D 2540 PUNCH B
SPOOL 00E 1403 A
DEDICATE 150 TSTRES
DEDICATE 191 TSTPK1
SPECIAL 080 2702 IBM
SPECIAL 081 2702 IBM
SPECIAL 082 2702 IBM
SPECIAL 083 2702 IBM
SPECIAL 070 3270
  
```

Figure 1-8. Directory Entry for a Multiple-Access Virtual Machine Running VM/SP under VM/SP

To connect a terminal supported by both VM/SP and a multiple-access system, use the CP DIAL command. Such terminals can be on either non-switched or switched lines. To connect a terminal to the virtual machine defined in Figure 1-8, enter this command:

```
dial testvm
```

The VM/SP system matches the terminal type to an equivalent virtual line that is available and enabled (see Figure 1-8: 070, 080, 081, 082, or 083). Once a

connection is made, the virtual machine controls the terminal to which it is logically connected (in this example the VM/SP virtual machine). It remains connected until one of the following happens:

- The user logs off using standard logoff procedures
- The virtual machine is forcibly logged off.
- The user issues a CP RESET command.
- The user issues a CP SYSTEM RESET command.
- The user issues a CP SYSTEM CLEAR command.
- The user issues a CP IPL command.

Once logged off, the user is then free either to logon to VM/SP or to use the DIAL command to contact another multiple-access system.

Dial-up terminals supported by a multiple-access system may be of a different type than those supported by VM/SP as virtual system consoles. Such terminals must be on switched lines, and the CP DIAL command cannot be used. Users must dial the multiple-access system's telephone numbers directly.

As shown in Figure 1-9, a communications system can be tested by using multiple virtual machines in place of multiple real machines. For example, while there exists a single two-line 2701 on the real machine, the virtual 2701 units could each be defined as a one-line 2701.

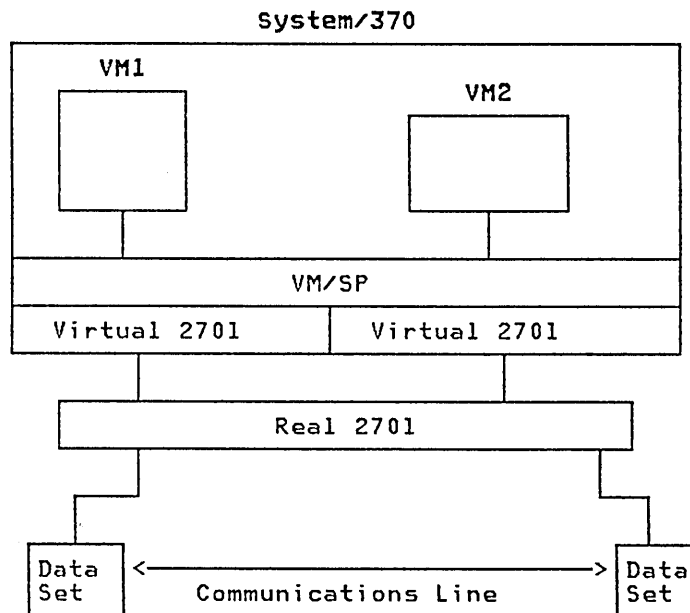


Figure 1-9. A Communications Test System

Note: This figure assumes that the real 2701 transmission control unit is equipped with the appropriate data sets and line capability.

Figure 1-10 illustrates a virtual transmission control unit running remote 3270 units.

When the terminals supported by the multiple-access system are not those supported by VM/SP as virtual operator consoles, the real line appearances must be one of the following:

- Defined in the VM/SP directory entry for the virtual machine via the DEDICATE control statement; for example:

DEDICATE **vaddr raddr**

where:

vaddr is the virtual address, and **raddr** is the real address of the appropriate line appearance on the real transmission control unit.

----- or -----

- Attached to the virtual machine by an operator with privilege class B; for example:

attach **raddr** to vm1 as **vaddr**

where:

raddr is the real address of the appropriate line appearance on the real transmission control unit, and **vaddr** is the address of the line appearance as generated in the virtual machine operating system.

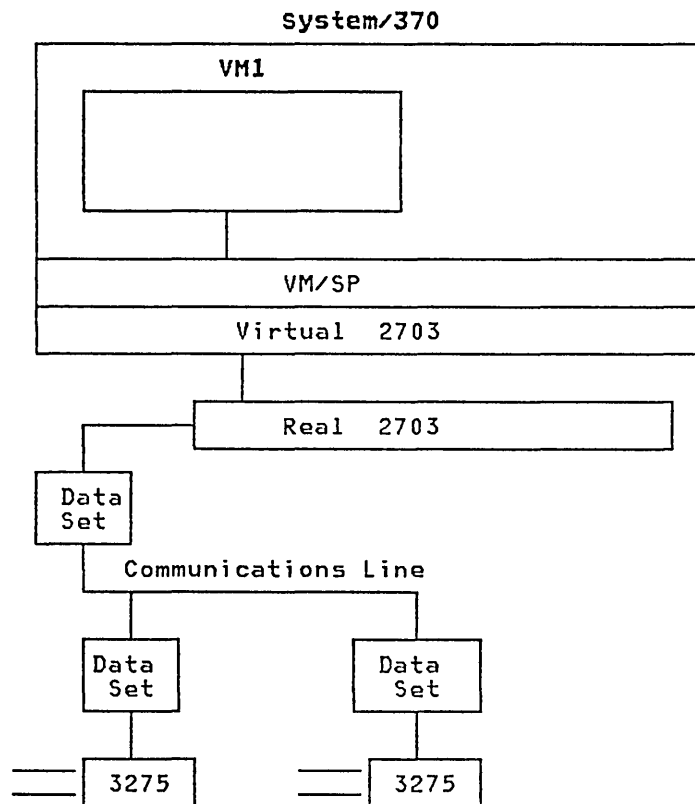


Figure 1-10. A Virtual 2703 TCU Controlling Remote 3270 Terminals

Performance Considerations

When virtual machine activity is initiated in an infrequent or irregular basis (such as from a remote terminal in a teleprocessing inquiry system), some (or all) of its virtual storage may be paged out before the virtual machine begins processing. The paging activity required for the virtual machine to respond to the teleprocessing request may increase the time required to respond to the request.

Use the locked pages or reserved page frames options to improve performance.

If the program must be run in the dynamic paging area, then locking specific pages of the virtual machine into real storage may ease the problem. However, besides page zero and the page containing the teleprocessing interruption handler, it is not always easy or possible to identify which specific pages are always required.

A more flexible approach than locked pages is the reserved page frames option. When a temporarily inactive virtual machine having this option is reactivated, these page frames are immediately available. If the program code or data required to satisfy the request was in real storage at the time the virtual machine became inactive, no paging activity is required for the virtual machine to respond.

For details about the locked pages and reserved page frames options, refer to the *VM/SP System Programmer's Guide*.

Shadow Table Maintenance Support

Shadow table maintenance support reduces the overhead associated with maintaining shadow page and segment tables. This support includes the following four areas.

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

Multiple Shadow Table Support

Multiple shadow table support reduces the number of shadow tables that VM/SP has to purge when a guest operating system in a virtual machine dispatches a new address space. This support adds a segment table origin control block (STOBLOK) and the STMULTI n option to the SET command.

STOBLOK keeps all the information about each shadow segment table, and VM/SP maintains a queue of STOBLOK control blocks with the associated shadow tables for each guest EC mode virtual machine. Thus, each time a guest operating system dispatches a new address space via a LCTL instruction, VM/SP can locate the proper shadow table to be used. By issuing the SET STMULTI n command, a user can define how many shadow tables that VM/SP should maintain concurrently for each virtual machine. The maximum number is six, and the default is three. The actual number specified varies by the amount of free storage available. Each segment table for a 16 megabyte address space requires 1024 bytes of storage, plus space for the page tables. To display the STMULTI specifications, a user can issue the QUERY SET command.

Selective Invalidation

Selective invalidation is a standard function of shadow table maintenance support. It allows VM/SP to selectively invalidate a shadow page table entry when a page frame is stolen or released from a guest EC mode virtual machine. Selective invalidation *always* takes place below the high-water mark that is established with the SET STBYPASS *nnmnk* command. Selective invalidation of the shadow page tables entries occurs *above* the high-water mark only if virtual machine assist is off. Total invalidation of the shadow page table entries occurs above the high-water mark whenever a page frame is released or stolen. After the guest EC mode virtual machine causes a page fault, virtual machine assist revalidates those entries above the high-water mark.

Elimination of One-Megabyte Shadow Tables

3081 processors do not permit use of one-megabyte segments for virtual machines. CP does not build shadow tables nor dispatch a virtual machine that uses 1Mb segments. On 3081 processors, virtual machines are restricted to 64K segments. Any attempt by a relocatable virtual machine using 1 Mb segments to use the DAT facility for address translation, results in a translation exception.

Shadow Table Bypass for V=R Users

Shadow table bypass for V=R users eliminates shadow tables for guest operating systems executing in the V=R virtual machine. By issuing the SET STBYPASS VR command, a user eliminates shadow tables and the overhead associated with maintaining them. VM/SP modifies the virtual operating system's page table to relocate virtual page zero to the highest real address within the V=R area. This relocation makes it possible for VM/SP to dispatch the virtual machine and have real control register 1 point to the guest page and segment tables.

Note: Single processor mode requires the use of shadow tables to simulate virtual prefixing. The SET STBYPASS VR command is ignored if issued. However, the SET STBYPASS *nnmnk* command is valid and should be used in the single processor mode environment.

SET STMULTI Command

If the SET STBYPASS VR command has been issued and shadow tables have been eliminated, the SET STMULTI command has no effect on the V=R guest virtual machine. However, the single processor mode V=R user running a guest AP or MP system can effectively use the SET STMULTI command.

Restrictions when Eliminating Shadow Tables

When shadow tables are eliminated, the following restrictions apply:

- The virtual system's real page zero must map only to its virtual page zero. Otherwise, STBYPASS VR will be set off and shadow tables will be used instead.
- No virtual machine segment or page tables can start in a relocated page. This means that VM/SP control register 1 and the segment table entries cannot point to the first 4K of storage.

- The system *cannot* use the relocated page table entry in these ways:

By looking at its contents

----- or -----

By executing a load real address (LRA) instruction on virtual page zero (normally mapped to real page zero), except for using the condition code returned by the instruction.

- The virtual operating system must have only one page table entry for its real page zero. If multiple address spaces are used, the page table must be shared by each address space that uses real page zero.
- Any dump taken of the virtual operating system may contain a relocated page table entry for page zero. Thus, any program designed to automatically read and interpret dumps must handle this condition.

Note: Once relocated by the SET STBYPASS VR command, the virtual operating system must continue to use the relocated page table entry without changing its contents or moving its location.

Shadow Table Bypass for V=V Users

V=V users executing SVS, MVS, or VM/SP under VM/SP can use shadow table bypass. This function allows V=V users to establish a common nucleus for multiple address spaces. By specifying the maximum size of the nucleus, a user reduces purge and invalidation time. Thus, when the virtual machine executes a PTLB or LCTL instruction, VM/SP invalidates or purges only the shadow table entries that are above the high-water mark (or highest limit) of the virtual operating system's nucleus that is mapped guest virtual=guest real.

SET STBYPASS Command

By using the SET STBYPASS *nnnnk* command (in conjunction with the STFIRST directory option), a user can define the high-water mark (or highest limit) of the virtual operating system's nucleus that is mapped guest virtual=guest real. This specification reduces the overhead associated with maintaining shadow page and segment tables VM/SP can selectively invalidate shadow table entries associated with pages stolen from within this V=R area.

The SET STBYPASS *nnnnK* (or *nnM*) specification can be either approximate or precise. Generally, the approximate value is sufficient. However, address spaces may abend with address space errors. When these errors occur, then specify a precise value.

To determine an approximate *nnnnk* (or *nnM*) value:

1. Issue the SET STBYPASS *nnnnK* command with *nnnnK* equal to the virtual machine storage size. This specification causes VM/SP to respond with the highest allowable high-water mark.
2. Reissue the SET STBYPASS *nnnnK* command with *nnnnK* equal to the highest allowable high-water mark from step 1.

Note: The highest allowable high-water mark may not be the true value. The virtual translation tables may, by chance, have several pageable page

frames in the V=R storage area that map contiguously with the true high-water mark. If this is the case, address spaces may abend with address space errors. When these errors occur, use one of the following methods to determine the precise *nnnnnK* value.

To determine the precise *nnnnnK* (or *nnM*) value follow one of these methods:

1. Locate entry CVTPVTP (X'164') in the SVS or MVS CVT. This is the address of the PVT.
2. Locate entry PVTFPFN (X'10') in the PVT. This is a half word value that represents the relative block number (RBN) of the first page frame table entry (PFTE) in the page frame table (PFT). (PVTFPFN is located at X'18' for the MVS/SP 1.3 or later guest virtual machine.)
3. The value at PVTFPFN is left justified and the 12 high order bits are the high order bits of a 24 bit address. Thus a value of X'0960' at PVTFPFN becomes X'096XXX' in an address. The 12 low order bits are zeroes, so the result is X'096000' for the address value you are looking for.

For the MVS/SP 1.3 or later guest virtual machine, the address calculation is different. The value at PVTFPFNs in this case is right justified and its 12 low order bits are the 12 high order bits of a 24 bit address. For example, if PVTFPFN contains a value of X'0096', drop the first 4 bits (the first zero) and begin the 24 bit address with PVTFPFN's last 12 bits. The address is now X'096XXX'. The 12 low order bits are zeroes so the resulting address is X'096000'.

4. Take the X'096000' and convert it to decimal. The result is 614,400.
5. Divide the decimal value 614,400 by 1024. The result will be the *nnnnnK* value you are looking for, in this case 600K.

Notes:

1. For virtual operating systems that use only one address space (OS/VS1, VS1/BPE, DOS/VS, AF-DOS/VS DOS/VSE, VSE/AF), severe performance degradation can be avoided on the 168-3, 3032, and 3033 processors by issuing the STBYPASS command with a high-water mark equal to the size of the virtual machine. This forces CP to dispatch the virtual machine as if it were running with dynamic address translation (DAT) off, no matter what the setting is in the virtual PSW. With DAT turned off, the virtual operating system is dispatched with 4K pages even though this operating system normally uses 2K page sizes. The use of 2K page virtual storage sizes results in a slower instruction rate because the high speed buffer is split in half and reset any time control register zero changes page sizes.
2. The nearer the value for *nnnnnK* (or *nnM*) to the virtual machine size, the greater the reduction in VM/SP overhead.
3. Also, the STBYPASS *nnnnnK* command should be used with single processor mode in AP and MP systems.
4. The STBYPASS command should not be issued until the operating system MVS or SVS) in the virtual machine has completed its initialization.

For more details about specifying the STBYPASS command, refer to *VM/SP CP Command Reference for General Users*.

To display the STBYPASS specifications, you can issue the QUERY SET command.

Restrictions when using Shadow Table Bypass V=V Users

When using shadow table bypass for the V=V user, the following restrictions apply:

- Below the high-water mark, the virtual operating system must map each virtual address, starting from location zero, to its real address.
- When multiple segment tables are used by a virtual operating system, the page tables that correspond to the area below the high-water mark must be common to all segment tables.
- To *invalidate* entries below the high-water mark, the virtual operating system must specify DIAGNOSE code X'10' to release the virtual pages. However, the operating system is not restricted when validating entries below this mark.
- After setting shadow table bypass for the V=V user, the shadow tables are initially invalidated and rebuilt. The shadow table bypass should be set before using the STMULTI command, otherwise the STMULTI command will be reset by setting the shadow table bypass.

STFIRST Directory Option

If virtual machine assist is available on the real machine, the STFIRST option must be in the virtual machine's directory to authorize the use of the SET STBYPASS command. Users can specify the STFIRST performance option in the OPTION directory control statement.

STFIRST should only be used for virtual machines that execute debugged and tested production workloads. STFIRST should not be specified for virtual machines executing test systems or programs that do not follow the programming restrictions for shadow table bypass. (These restrictions are listed under the topic "Shadow Table Maintenance Support" in this chapter.) Otherwise, either VM/SP could abnormally terminate or some other unpredictable result could occur.

SET STMULTI Command

When a virtual operating system uses more than one segment table, a user can issue the SET STMULTI command to define:

- How many shadow tables (maximum of 6) that CP is to support for the virtual machine -- n operand.
- The number of contiguous shadow page tables (in each pool of shadow page tables) for the virtual operating system's dynamic paging area -- USEG xx operand (SVS and MVS users only). USEG xx can be set to zero or can range from 8 through 99.
- The number of contiguous segments for the common area (at the high end of an address space) that is shared by all address spaces within the virtual operating system -- CSEG yyy operand (SVS and MVS users only).

Note: The purpose of the USEG and CSEG parameters is to improve CP performance by decreasing the number of shadow page tables that CP must maintain. Furthermore, USEG must be specified in order to specify CSEG. If these values are too low or the USEG parameter is specified without the CSEG parameter, poor virtual machine performance can result. Also, you can turn off the USEG or CSEG performance options by specifying a zero in either parameter. However, a zero value nullifies the performance benefits.

To display the STMULTI specifications, the user can issue the QUERY SET command.

Before using the STMULTI command, determine the values to specify on the command. The following description explains one method for calculating these values.

The n Operand: Define this value according to the operating system used in the virtual machine:

- For SVS, specify two.
- For MVS, specify a value equal to the average number of initiators that are active at one time, plus two (to equal one address space for the nucleus and one address space for the master scheduler). (However, unless the value in field DMKSYSMS in module DMKSYS is also changed, the value cannot exceed the maximum of six.)

Because shadow table maintenance support limits the maximum number of STOBLOKs (segment table origin control blocks) to six, the actual number used depends upon the number of active MVS address spaces. If the demand for STOBLOKs exceeds the maximum number allowed, considerable STOBLOK stealing may take place and degrade the virtual machine's performance.

To monitor STOBLOK stealing, use the VM/SP Monitor. It collects these statistics about shadow tables in the class 4 code 1 record: (1) the maximum number allowed (the n operand), and (2) the actual number of active address spaces. To reduce these statistics and print them, use the VM/370 Performance/Monitor Analysis Program (VMAP), 5798-CPX.

The USEG xx Operand: Define this value based on the value of the page table steal counter in the ECBLOK (extension to VMBLOK for virtual machine with relocate).

By experimenting with different USEG values over various time periods, users can calculate an appropriate value for their operating systems. For example, when the increase in the counter value is high (a three- or four-digit hexadecimal value), increase the USEG value. When the increase in the counter value is low (a two-digit hexadecimal value), the USEG value is probably appropriate.

To automatically monitor the value of the page table steal counter, use the VM/SP Monitor. It periodically collects the value of this counter and other shadow table maintenance counters in the class 4 code 1 record. To reduce these statistics and print them, use the VM/370 Performance/Monitor Analysis Program (VMAP), 5798-CPX.

To manually locate the page table steal counter, follow these instructions:

1. Enter (or have entered) this CP command (class C and E):

```
#cp loc userid
```

The userid in the LOCATE command is the name of the user's virtual machine. The command prints the address of this user's virtual machine block (VMBLOK).

2. Add X'0C' to the VMBLOK address to locate the pointer to the ECBLOK (VMECEXT field).
3. Locate the ECBLOK.
4. Add X'7C' to the ECBLOK address to locate the page table steal counter (EXTUPTST field).

The CSEG yyy Operand: Define the yyy value to equal the number of 64K segments in the SVS or MVS common area. The calculation below represents the maximum size for the common area. To calculate this value follow these steps:

1. Run the AMBLIST service aid program to find the beginning address of the PLPA.
2. Subtract the address found in step 1 from X'FFFFFF' and convert it from hexadecimal to decimal.
3. Divide the result from step 2 by 65,536 (64K) and round it to the nearest 64K segment.
4. Specify the decimal value in step 3 in the CSEG operand.

However, specifying this size for the CSEG operand may not provide the best performance. To obtain better performance, organize the SVS or MVS PLPA by packing frequently used modules together and putting them in the high address range of the PLPA. Then, subtract the size of this packed area from the maximum PLPA size. This CSEG value should represent the PLPA size from its beginning address to the lower address boundary of the packed area. To calculate this value:

1. Locate entry CVTSHRVM (X'1A0') in the SVS or MVS CVT.
2. Subtract the value in entry CVTSHRVM from X'FFFFFF' and convert the result to decimal.
3. Divide the result from step 2 by 65,536 (64K) and round it to the nearest 64K segment.
4. Specify the decimal value in step 3 in the CSEG operand.

Note: For better performance in single processor mode, use a CSEG value that represents the maximum size of the common area.

Performance Guidelines

When run in a virtual machine, the performance characteristics of an operating system are difficult to predict. This unpredictability is a result of the complex interaction of many factors that affect performance. These factors can be broadly classified into three groups:

- Configuration factors
- Workload factors
- VM/SP performance factors

Performance of any virtual machine may be improved by the choice of hardware configuration, operating system workload, and VM/SP performance options. While a specific virtual machine's performance may not equal that of the same operating system running stand-alone on the same System/370, in some situations the total throughput obtained in the virtual machine environment can be equal to, or better than, that obtained on a real machine.

Configuration Factors Influencing Performance

These hardware configuration factors influence the performance of an operating system in a virtual machine:

- The System/370 model used.
- The amount of real storage available.
- The speed, capacity, and number of paging devices.
- The degree of channel and control unit contention, as well as arm contention affecting each paging device.
- Whether virtual machine assist or VM/370 extended control-program support is installed on the hardware and enabled.
- Interference between system paging devices and devices for processing a user's I/O requests.

When discussing these performance factors, this discussion assumes that the reader is familiar with the need to design an optimal configuration for a specific workload and operating system.

When moving a specific workload and operating system to the virtual machine environment, an installation should plan for an increased need in such hardware requirements as real storage, DASD space, and processor speed. While VM/SP's overhead for dispatching, scheduling, and paging is relatively small, the overhead for simulating privileged instructions may be considerable.

When not operating under VM/SP, an operating system runs directly on its own hardware (native mode) and manages its resources through the use of privileged instructions (such as SIO and LPSW) issued in supervisor state. When executing in a virtual machine, VM/SP dispatches the operating system in problem state, and any privileged instructions issued by the virtual machine causes a real privileged instruction exception interruption. This interruption transfers control to VM/SP to simulate the instruction. The amount of work done by VM/SP in analyzing and handling a virtual machine-initiated interruption depends upon the type and

complexity of the interruption. Thus, any reduction in the number of privileged instructions issued by a virtual machine's operating system reduces the amount of extra work VM/SP must do to support that operating system.

Note: Virtual machine assist support has been specifically designed to reduce VM/SP's overhead associated with simulating privileged instructions. It is the most effective method for reducing privileged instruction simulation. Any installation that is going to run a production operating system under VM/SP should consider virtual machine assist as a prerequisite for improving performance. Other steps for improving performance (such as using specialized VM/SP performance functions) are of secondary importance compared to using virtual machine assist.

VM/370 extended control-program support (ECPS: VM/370) is a hardware assist function that provides support over and above that provided by virtual machine assist. It improves VM/SP performance by reducing VM/SP's real supervisor state time needed to support virtual machines. The *VM/SP System Programmer's Guide* lists the specific functions of ECPS: VM/370 that certain System/370 models support.

Workload Factors Influencing Performance

These workload factors influence the performance of an operating system in a virtual machine:

- The type of operating system being used.
- The total number of virtual machines running under VM/SP.
- The type of work each virtual machine is doing, especially the amount of I/O processing required.

By measuring and evaluating the effects of these workload factors on a specific configuration, an installation can understand their effect on performance.

To relate these measurement values to system workload for a specific configuration, an installation must define its workload. The definition of workload varies with the environment:

<i>Environment</i>	<i>Workload Definition</i>
Interactive time-sharing	Arrival rate of transactions and the processor time and working set size required for each transaction.
Batch system	Job throughput and resource requirements (processor time, region or partition size, and number of SIOs issued) for each job.

By using these workload definitions, an installation can measure its workload under VM/SP as follows:

<i>Environment</i>	<i>Workload Measurement</i>
CMS under VM/SP	Number of active users
Operating system under VM/SP	User I/O requests executed

When both an operating system and CMS run under the same VM/SP system, workload measurement depends upon which type of environment dominates the VM/SP system.

To measure workload performance in a specific configuration, you can use the Field Developed Program VM/SP Performance/Monitor Analysis program (5798-CPX). This program plots a number of important system variables (such as processor usage, various contention measurements, and paging rates) against workload measurement for both the CMS and operating system workloads under VM/SP. For a specific configuration, it allows you to relate processor usage, storage usage, and resource contention to the total system workload in both interactive and batch production environments.

By using this analysis program, you can eventually determine the optimum processor model, storage size, and I/O configuration for a specific workload. You may determine that you need to do such things as: redistribute data sets to reduce arm contention, add control units and channels to reduce I/O contention, and add paging devices to reduce interference between system and user I/O processing.

VM/SP Performance Factors

These specialized VM/SP software techniques influence the performance of an operating system in a virtual machine:

- Whether VM/VS handshaking is used.
- The type and number of VM/SP performance options in use by one or more virtual machines.

VM/VS Handshaking: VM/VS handshaking (described earlier in this section under the topic “VM/VS Handshaking”) permits duplicate processing between CP and the guest operating system to be held to a minimum. It also permits VM/SP to simulate privileged instructions.

VM/SP Performance Options: After measuring the performance of both VM/SP and the virtual machines it supports, the system analyst and the general user can each use certain VM/SP performance options. These options allow them to create a special performance environment for one or more virtual machines. The options allow:

- The system programmer to redistribute system resources either to balance them or to favor one virtual machine over another.
- The general user to improve the performance for his virtual machine.

The VM/SP system operator, or system programmer, can give certain options to a specific virtual machine to improve its performance over other virtual machines. A general user has certain performance options that give limited control over his virtual machine. The options available to the system operator and the general user are:

<i>System Operator</i>	<i>General User</i>
Locked pages option	Virtual machine assist
Reserved page frames	VM/370 extended control-program support
Priority	Virtual=Real option ²
Favored execution option	STBYPASS command for a virtual machine
QDROP OFF	

The following options can be applied to only one virtual machine at a time.

- Reserved page frames
- Virtual=Real option²

The following options can be applied to as many virtual machines as desired:

- Favored execution with guaranteed percentage
- Basic favored execution (without guaranteed percentage)
- Priority
- Virtual machine assist
- VM/370 extended control-program support (ECPS: VM/370)
- Locked pages
- QDROP

For basic information about these options, refer to the *VM/SP System Programmer's Guide*. For details about specifying the options for the system operator, refer to the *VM/SP Operator's Guide*. For details about specifying the options for the general user, refer to the *VM/SP CP Command Reference for General Users*.

The following performance-related additions to the VM/SP system control program are available. For certain environments, these additions:

- Improve throughput
- Provide better terminal response
- Reduce paging overhead by keeping the most active pages on preferred DASD, and migrating the inactive pages to slower devices
- Reduce overhead associated with maintaining shadow page and segment tables

² This option cannot be specified in a command. To obtain it, a general user requests the VM/SP system administrator to specify it on the OPTION control statement (VIRT=REAL option) for the user's virtual machine directory entry. The CP nucleus must also be generated with the V=R option.

- Improve performance for a production virtual storage operating system running under VM/SP
- Increase throughput of MVS running under VM/SP on an attached processor or multiprocessor system
- Increase throughput of MVS running under VM/SP on the appropriate processor by using VM and MVS microcode ASSIST concurrently.

Performance Measurements

Performance measurements apply to both the VM/SP system and the individual virtual machine. How well the system responds is of prime importance to the general user. How efficiently the individual virtual machine makes use of the allotted storage, processor, and I/O facilities is of prime importance to the system analyst.

VM/SP provides certain CP commands (INDICATE and MONITOR) that allow both VM/SP and virtual machine performance to be tracked and measured; other commands allow the setting of certain options to improve performance. To reduce and help analyze the data produced by the MONITOR command, the Field Developed Program VM/370 Performance/Monitor Analysis program (5798-CPX) is available. By using this program, an installation can eventually determine its optimum processor model, storage size, and I/O configuration for a specific workload. For a complete description of the INDICATE and MONITOR commands, refer to the *VM/SP System Programmer's Guide*.

Emphasizing Interactive Response Times

Most conditions for good performance established for the time-sharing and batch systems apply equally well to mixed mode systems. However, two major factors make any determination more difficult to make. First, get evidence to show that, in all circumstances, priority is given to maintaining good interactive response, and that nontrivial tasks really execute in the background. Second, background tasks (no matter how large, inefficient, or demanding) should not be allowed to dominate the overall use of the time-sharing system. In other words, in mixed mode operation, get evidence to show that users with poor characteristics are discriminated against for the sake of maintaining an efficient system for the remaining users.

A number of other conditions are more obvious. For example, measure response time and determine at what point it becomes unacceptable and why. Studies of time-sharing systems have shown that a user's work rate is closely correlated with the system response. When the system responds quickly, the user is alert, ready for the next interaction, and thought processes are uninterrupted. When the system response time is poor, the user loses concentration.

Generation Procedures Under VM/SP

VM/SP can help considerably throughout the guest system generation process. Probably VM/SP's biggest advantage is the ability to generate the system under VM/SP without disturbing the normal production activity.

The system programmer (or whoever is responsible for the guest operating system) can log on to his own virtual machine and go through the generation steps at his own pace while the daily work is being processed. He can use the System Product Editor to create and update the job streams that are used during system generation.

Whenever the system generation process requires, he can use EXEC procedures to pass these saved job streams to the test system. When the system is tested, it can be placed online, replacing the previous version with minimal interruption to the production activity.

For a discussion of the System Product Editor refer to the *VM/SP System Product Editor User's Guide* and the *VM/SP System Product Editor Command and Macro Reference*. The System Product Interpreter is described in the *VM/SP System Product Interpreter Reference*. The EXEC 2 facility is described in the *VM/SP EXEC2 Reference*. For details about the system generation procedures for DOS and OS, refer to the appropriate operating system libraries.

Creating VM/SP Directory Entries

To allow a virtual machine to exist in the VM/SP system, the VM/SP system requires a directory entry definition. Each definition is kept in a directory entry source file on a user minidisk. An installation must use the VM/SP directory program to convert these source definitions in the VM/SP system directory file (usually on the system residence disk) that contains one entry for each virtual machine.

Each directory entry contains a number of directory control statements that define the virtual machine's configuration and other operational characteristics to VM/SP. In general, a virtual machine configuration defined in the directory consists of the following:

- Virtual storage, console, and processor
- Direct access storage devices
- Unit record devices
- Other devices

Figure 1-11 on page 1-51 shows the relationship of a directory to both the VM/SP system's real devices and the virtual machine's virtual devices. You must keep both the source and system directories updated. As users submit additions and/or changes, you must either create new or update current directory entries. This updating can be done by using the VM/370 Directory Maintenance Program Product (5748-XE4), the System Product Editor, or punched cards. (For more details about this program product, refer to the *VM/370 Directory Maintenance Program Product General Information Manual*, GC20-1836.)

To create directory entries for operating systems running in virtual machines, you must consider both the general and unique requirements for specifying directory entries. For general details about specifying directory entries, refer to the *VM/SP Planning Guide and Reference*. For more details about specifying directory entries for operating systems running in a virtual machine, refer to the following topic "Unique Directory Entry Considerations."

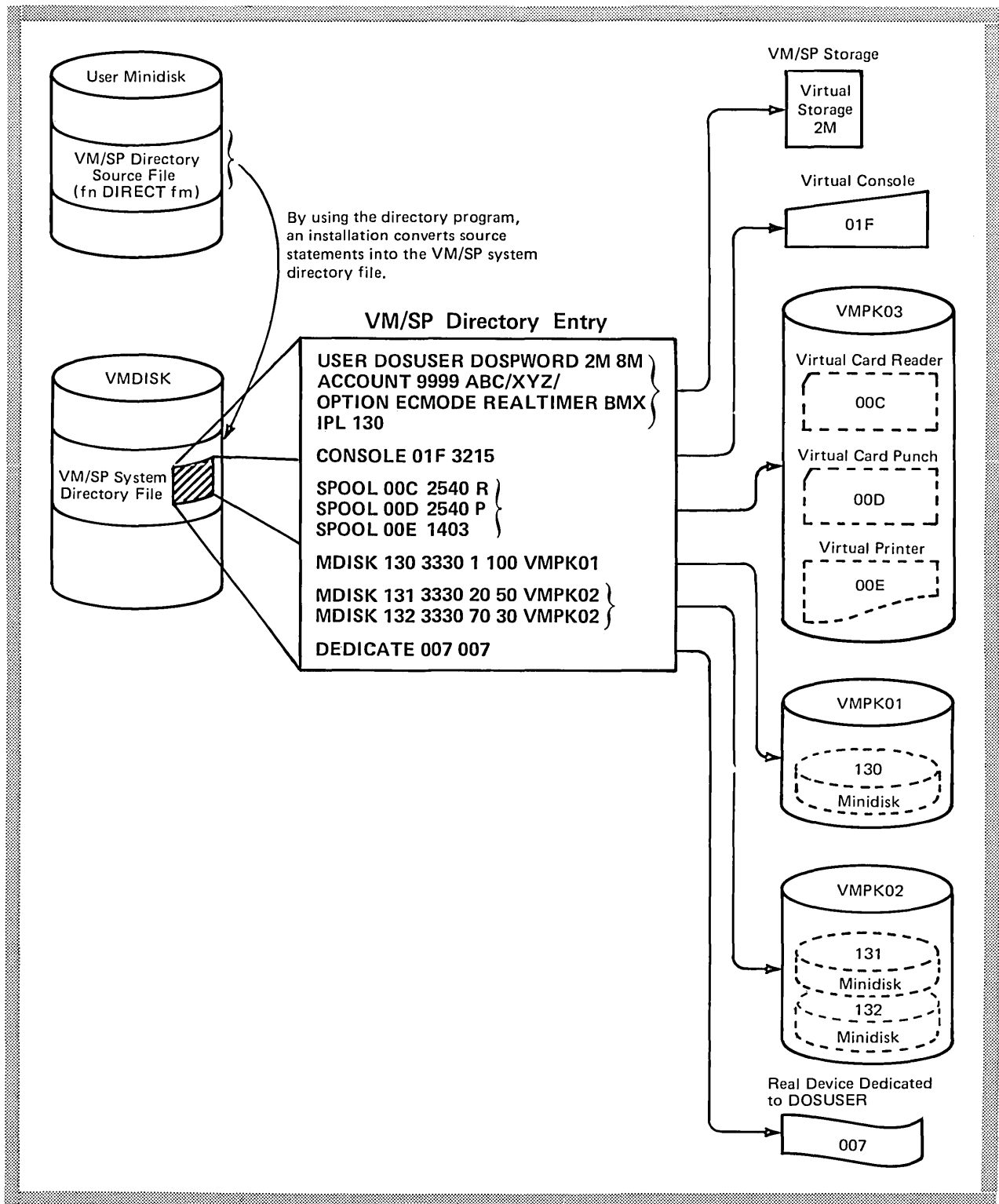


Figure 1-11. Relationship of VM/SP Directory Entries

Unique Directory,Entry Considerations

This topic lists directory control statements with unique considerations for running an operating system in a virtual machine.

USER Control Statement

The USER control statement has no unique considerations for running an operating system in a virtual machine.

ACCOUNT Control Statement

The ACCOUNT control statement has no unique considerations for running an operating system in a virtual machine.

OPTION Control Statement

See "Virtual Machine Options" discussed earlier in this section.

IPL Control Statement

If a virtual machine runs one operating system most of the time, the users can have that system automatically loaded every time they log on. Use the IPL statement to identify the operating system to VM/SP, such as:

```
ipl 130
```

Virtual address 130 represents the address of the device that contains the system to be loaded. If the virtual machine's operating system has been "saved" (by using the CP SAVESYS command), specify:

```
ipl dosvs
```

DOSVS is the name under which the system was saved.

Note: For the VM/SP system operator to automatically log on to a virtual machine (by using the class A or B AUTOLOG command), the virtual machine's directory entry must contain an IPL control statement.

CONSOLE Control Statement

If you specify 3270 on the console control statement, you can alternate between 3215 mode for CP commands and 3270 full screen mode for guest operating systems. However, a secondary userid must be specified in this directory statement whenever you want to use the Single Console Image Facility. The secondary userid denotes another user that controls all messages, replies and commands for a virtual machine after the primary user disconnects. This gives an operator added flexibility in an environment where service virtual machines are used because the operator can control several disconnected machines (via CP SEND command) from one physical terminal.

MDISK Control Statement

If you are running VSE/AF release 2 or 3 with shared spool and/or VSAM via DOS's lock files, specify "V" when using the MDISK control statement with your other read/write options.

SPOOL Control Statement

The SPOOL control statement supports the specification of a virtual printer for use by the virtual machine.

DEDICATE Control Statement

Use the DEDICATE control statement to provide a virtual machine with a corresponding real device. The virtual machine has sole use of the dedicated device.

Magnetic Tapes: A device such as a magnetic tape drive can be used by only one virtual machine at a time; therefore, specify it in a directory entry with a DEDICATE statement. For example:

```
DEDICATE 181 281
```

This statement allows the operating system to access the device at real address 281 via a virtual address of 181.

Unit Record Devices: In many cases, spooling represents the most efficient way of handling the unit record input and output of many virtual machines. However, special cases may justify the dedication of a real unit record device to a single virtual machine.

One special case is when the virtual machine's operating system does its own spooling, such as VSE/POWER under VSE/AF or JES under MVS/SP. To eliminate double spooling of printer output, include a DEDICATE statement in the virtual machine's directory entry, such as:

```
DEDICATE 00E 002
```

This statement causes VM/SP to pass all virtual printer 00E output directly to the real printer at 002.

Note: The operating system must support the real printer used. For example, DOS/VS Release 34 does not support the 3203 model 5, therefore CP must support that printer and the DOS virtual machine directory must have a spool statement for supported printers.

Another case where a user may want a unit record device dedicated to a virtual machine would be if the virtual machine produced a sufficient volume of output to keep the device busy.

Users can also have the system operator dynamically dedicate a unit record device to their virtual machine. Send the system operator the message:

```
#cp msg operator Please attach real punch 00d to me as 00d
```

Assuming that the real punch at 00d is not in use by the system or any other virtual machines, the operator responds:

```
ATTACH 00d TO USERID AS 00d
```

When the device is attached, VM/SP sends a confirmation message to *userid*:

```
PUN 00D ATTACHED
```

When the device is no longer needed, the user can issue the detach command.

Send the operator a message when the device is no longer needed.

```
#cp msg operator Thanks, I'm done with punch 00d.
```

Remote Devices: The DEDICATE statement can be used to attach remote 3270 Information Display System Printers (3284, 3286, 3287, and 3289) to a virtual machine. For example, a directory entry can include the statement:

```
DEDICATE NETWORK 00E 0102
```

00E is the virtual address of the device in the virtual machine and 0102 is the resource ID as specified in DMKRIO. Remote 3270 Information Display System Printers can also be attached by the NETWORK ATTACH command. For more details, see the *VM/SP Operator's Guide*.

Unsupported Devices: The DEDICATE statement can be used to place a device that VM/SP does not support into a virtual machine's configuration. To dedicate a device, the device must:

- Be physically connected to the System/370
- Be supported by the virtual machine's operating system
- Not violate any of the restrictions contained in the VM/SP restriction section of the *VM/SP Planning Guide and Reference*.

For example, a directory entry can include the statement:

```
DEDICATE 007 012
```

Where real address 012 could represent a 2671 Paper Tape Reader that is part of the System/370 on which VM/SP is running. If the operating system was generated with a 2671 defined at address 007, VM/SP handles the device and CCW address translation associated with reading from the device. The operating system in the virtual machine is responsible for error recovery and error recording procedures.

2305 Devices: When using the DEDICATE statement to attach a 2305 to a virtual machine, both the real and virtual addresses must refer to the first base device address on the unit. The first base address of the 2305 is 0 or 8 -- the resulting address appears as xx0 or xx8. However, when VM/SP processes the statement it creates all eight addresses (0-7 or 8-F) for the 2305.

LINK Control Statement

The LINK control statement has no unique considerations for running an operating system in a virtual machine.

SPECIAL Control Statement

Use the SPECIAL control statement to add I/O devices that do not require corresponding real devices. Some examples are:

- Virtual consoles
- Virtual communication line
- Virtual Channel-to-channel devices
- Pseudo timers
- Communication lines.

You can use the SPECIAL control statement to specify a virtual transmission control unit for a multiple-access operating system (such as MVS/TSO and CICS). For example, if the system requires three communication lines from a 2703, specify:

```
SPEcial 061 2703 IBM Tele
SPEcial 062 2703 IBM Tele
SPEcial 063 2703 IBM Tele
```

Before a terminal can communicate with a multiple-access system, the terminal user must issue the DIAL command to connect to any available line port.

```
dial userid
```

To connect to a particular line issue:

```
dial userid 062
```

Note: Of the three SPECIAL control statements specified in the preceding example, one was a teletypewriter line and two were IBM terminal lines. When the DIAL command is issued with no specific address, VM/SP connects the terminal to any available line as defined in the the SPECIAL control statement; the line then belongs to the specified userid. If no lines are available or if all lines are busy, VM/SP issues an error message and does not make the connection. To drop a dialed line, the operator of the multi-access virtual machine must issue the CP RESET command for that terminal's virtual address. Installations with post Release 2 can now power on/off to drop the dialed connection.

Defining Virtual Devices

When using the SPOOL, DEDICATE, and SPECIAL control statements to define virtual devices, the rules for assigning virtual addresses are the same as for real devices and control units. The type of subchannel required by a device's control unit dictates the valid address assignments. Devices that need special I/O interface protocol from control units, such as shared subchannels, require that all 16 device addresses (0-F) be reserved. Therefore, you can only attach similar devices to a control unit with a shared subchannel. For devices that need a control unit with a nonshared subchannel, only one address per device is required. Devices that attach to a control unit with a nonshared subchannel do not have to be the same type.

For example, if the directory entry specifies:

```
SPOOL 102 3211
SPECIAL 103 3270
```


The 3270 specified at address 103 requires a shared subchannel and therefore reserves addresses 100-10F for display type devices. Since the device specified at address 102 is not a display unit but a printer, processing of channel programs involving these two devices can result in a hung or busy condition.

AUTOLOG Facility

AUTOLOG is a convenient way to initiate large production operating systems with many I/O devices that run under VM/SP. The I/O devices needed by these operating systems require considerable contiguous storage space for the I/O control blocks established by VM/SP. If smaller users have logged onto VM/SP before these large operating systems are started, there may not be sufficient contiguous storage space available for the required I/O control blocks. The logon of the virtual machine will still be completed even if the I/O control blocks can not be established. Therefore, there may be an insufficient number of I/O devices to run the operating system and its application programs.

To ensure sufficient contiguous storage space, log on to the large production machines immediately after loading VM/SP.

- Have the VM/SP system operator issue the CP AUTOLOG command before enabling user terminals.

----- or -----

- Define the AUTOLOG1 virtual machine in the VM/SP directory. The AUTOLOG1 virtual machine is automatically logged on immediately after VM is loaded and can be used to logon and load virtual machines that require substantial contiguous storage.

Using the CP AUTOLOG Command

Before enabling user terminals, the VM/SP system operator can issue the CP AUTOLOG command for each production virtual machine that requires substantial contiguous storage. The directory entry for the userid indicated by the CP AUTOLOG command must contain an IPL statement for the desired operating system. For more information about the CP AUTOLOG command, refer to the *VM/SP Operator's Guide*.

Defining AUTOLOG1 in the Directory

To use AUTOLOG1 to initiate several virtual machines, have the VM/SP directory statements load CMS for the AUTOLOG1 userid. Include one or more CP AUTOLOG command in the PROFILE EXEC. Each AUTOLOG command initiates one virtual machine containing the desired operating systems. When using the CP AUTOLOG command, the directory entries for the virtual machine referenced by the CP AUTOLOG command must contain an IPL statement.

As a result of the CP AUTOLOG command in the PROFILE EXEC, the virtual machine is loaded. The operating system user then gains access to the virtual machine by doing *one* of the following:

- By logging on with the userid specified in the CP AUTOLOG command
- By issuing the CP SEND command through the secondary user's console
- By issuing the CP DIAL command and specifying the guest userid.

When the user logs off, contiguous storage space is relinquished. If the user wants to keep the virtual machine's I/O blocks in contiguous storage and temporarily relinquish use of the virtual machine, the user issues the CP DISCONN command. To reestablish usage, the user issues the CP LOGON command to reconnect to the virtual machine.

Multiple Systems With AUTOLOG1

In the next figure the AUTOLOG1 initializes CMS in a virtual machine. The virtual machines containing the production operating systems are automatically logged on in disconnect mode from the PROFILE EXEC. The PROFILE EXEC contains several CP AUTOLOG commands; one for each virtual machine to be loaded. For each userid identified in a CP AUTOLOG command, there must be an IPL statement in the VM/SP directory to load the appropriate operating system into the virtual machine. The last CP command in the PROFILE EXEC may logoff AUTOLOG1. The virtual machines are logged onto VM/SP in disconnect mode.

```
AUTOLOG1 Virtual Machine

USER AUTOLOG1 PASSWORD 512K 1M ABG
ACCOUNT ACCTNO BIN1
IPL CMS
  CONSOLE 009 3215
  SPOOL 00C 2540 R
  SPOOL 00D 2540 P
  SPOOL 00E 1403
  LINK MAINT 190 190 RR
  LINK MAINT 19E 19E RR
  LINK MAINT 19D 19D RR
  MDISK 191 3330 1 1 UDISKA WR RPASS WPASS

PROFILE EXEC

/* PROFILE EXEC for AUTOLOGing virtual machine */
TRACE E; ADDRESS COMMAND;
CP SPOOL CONSOLE START; CP SET EMSG ON;
EXEC TELL OP Now AUTOLOGing on the Guest Operating Systems;
CP AUTOLOG DOSUSER PASSDOS;
CP AUTOLOG DOSVUSER PASSDOSV;
CP AUTOLOG OSUSER PASSOS;
EXIT;
```

Figure 1-12. AUTOLOG1 Virtual Machine and PROFILE EXEC

By having the preceding AUTOLOG1 directory entry and PROFILE EXEC, the DOSUSER, DOSVUSER, and OSUSER virtual machines (specified in the PROFILE EXEC) are now logged onto the VM/SP system in disconnect mode. A user accesses these virtual machines through their secondary user's consoles, if any, or by logging on with the userid of DOSUSER, DOSVUSER, or OSUSER along with the appropriate password. To temporarily relinquish use of one of these virtual machines without relinquishing contiguous storage, a user issues the CP DISCONN command. To reestablish use, a user issues the CP LOGON command. Issuing the CP LOGOFF command releases the contiguous storage space containing the VM/SP virtual I/O device control blocks.

Sample Directory Entries

This topic shows some virtual machines that can be defined when running operating systems in virtual machines. Sample directory entries for running specific operating systems under VM/SP are in the operating system sections of this book.

A Multiple-Access Virtual Machine

The following directory entry represents a multiple-access TSO system configured to handle one to four concurrent remote terminals and one local 3270. It has been given the VIRT=REAL option to improve response time.

```
USER TSOSYS PASSWORD 2M 4M G
ACCOUNT ACCTNO BIN8
IPL 290
OPTION REALTIMER VIRT=REAL ECMODE BMX 370E
CONSOLE 01F 3215
SPOOL 00C 2540 R
SPOOL 00D 2540 P
SPOOL 00E 1403
DEDICATE 290 TSOSYS
DEDICATE 291 TSOWRK
SPECIAL 070 3270
SPECIAL 080 2702 IBM TELE
SPECIAL 081 2702 IBM TELE
SPECIAL 082 2702 IBM TELE
SPECIAL 083 2702 IBM TELE
```

Figure 1-13. Sample Directory Entry for Multiple-Access TSO System

WHERE OPTION:

VIRT=REAL gives userid TSOSYS the capability to run in VM/SP's virtual=real area if it is available.

BMX allows CP to provide virtual block multiplexer channel services to the guest operating system.

370E allows userid TSOSYS to make use of MVS/System Extension Support.

DEDICATE defines the DASDs to be used by userid TSOSYS.

SPECIAL defines communication paths to userid TSOSYS for 3270 and printer/keyboard devices.

Summary

To run guest operating system in a virtual machine, you should:

- Use VM/VS Handshaking if possible.
- Eliminate double paging.
- Design new and existing applications to operate efficiently in the chosen paging environment.
- Use dual VM/MVS microcode assist if it is available on the processor when running MVS in a virtual machine.

When running specific multiprogramming operating systems under VM/SP (such as VSE/AF or MVS/SP), you should consider how that system interacts with VM/SP -- especially when that system has a page wait or I/O wait. To interact with these systems, VM/SP provides VM/VS handshaking for certain DOS and OS systems and the diagnose interface for guest operating systems.

Other areas to consider when running multiprogramming operating systems under VM/SP are:

- Spooling
- Channel model-dependencies
- Whether to use multiple or alternate consoles
- The states of virtual devices (dedicated, shared and spooled).

VM/SP also supports alternate paths, multiple-access virtual machines, operating systems using DASD reserve/release, and the ASP virtual machine. Installations can also alternate between operating systems under VM/SP.

While difficult to predict, performance of any virtual machine may be improved by the choice of hardware, operating system, and VM/SP options. VM/SP also provides the INDICATE and MONITOR commands to track and measure both VM/SP and virtual machine performance.

VM/SP can help considerably throughout the system generation process. Its biggest advantage is allowing an installation to generate a system under VM/SP without disturbing production activity.

To allow virtual machines to access the VM/SP system, the VM/SP system requires a file of directory entries that contains one entry for each virtual machine. Each directory entry contains a number of directory control statements that defines the virtual machine's configuration and operational characteristics to VM/SP. Some directory statements have unique considerations when running an operating system in a virtual machine. There is an AUTOLOG facility to automatically initiate large production operating systems with many I/O devices under VM/SP.

| Section 2. VM/SP in a Virtual Machine

Running VM under VM provides a convenient way to update and test VM without disturbing your production VM system. The system programmer can test new releases, isolated from any work currently running elsewhere in the system. The test system is the functional equivalent of a real processor and I/O devices.

The system programmer can test service updates, new configurations and modifications, and also train operators. Basically, there are two methods for running VM under VM: using minidisks or real disks. For the purpose of this discussion, our example uses minidisks.

| VM/SP Directory Definition

Throughout Section 2 the following terms will be used:

First level system = real CP system = reference to CP that is in real storage and that sees real device addresses.

Second level system = test CP system = reference to CP that is in a virtual machine.

To test a VM/SP system in a virtual machine you must create a directory entry for the test system. The test system's directory (a small and separate subset of the real system), need only specify the minimum number of users necessary to perform the testing. Make sure you define the test system's virtual machine large and varied enough to perform all necessary functions.

The following is a sample directory entry for a test system with userid TESTSYS. This directory allows most testing to be done from one userid rather than having several userids involved. It also has the options necessary to define a minimum system.

```
USER TESTSYS PASSWORD 4M 8M G
ACCOUNT NUMBER BIN11
OPTION ECMODE REALTIMER BMX
CONSOLE 01F 3215
SPOOL C 2540 READER
SPOOL D 2540 PUNCH
SPOOL E 1403
LINK MAINT 190 190 RR
LINK MAINT 19D 19D RR
LINK MAINT 19E 19E RR
MDISK 330 3330 1 50 SYSWRK WR RPASS WPASS
MDISK 331 3330 51 50 SYSWRK WR RPASS WPASS
```

Figure 2-1. VM under VM: Sample Directory Entry for TESTSYS

WHERE:

The USER statement defines the userid as TESTSYS, the password as PASSWORD, and 4M of storage (default) with 8M of storage as maximum. The actual storage size that you define for the test virtual machine should be equal to or

greater than the size of the real memory of the processor normally running this VM/SP system. For example, if the real VM/SP system runs in a 4M machine, the test system should be defined as 4M or larger.

The `OPTION` statement specifies the `ECMODE`, `REALTIMER` and `BMX` options. The `ECMODE` option is required so that the virtual machine can operate in extended control mode. The `REALTIMER` option causes the virtual machine to wait for a timer interruption to continue processing. The `BMX` (virtual block multiplexer) option allows an operating system running in a virtual machine to overlap multiple SIO requests on a specified channel path.

The `CONSOLE` and `SPOOL` statements specify the console and spool device addresses. These addresses must match the same addresses as the real system configuration. If that configuration is not used for the test system operation, they must match whatever configuration is specified in `DMKRIO` of the test system.

The `LINK` statements give this virtual machine access to CMS. Special considerations have to be taken in order to operate CMS. These considerations are described later in this section.

The `MDISK` statements for 330 and 331 define disks for the CP system residence, paging, and spooling volumes.

Notes:

1. This directory entry configuration does not define any other user disks, graphic devices, or tape drives. All additional devices required for testing VM/SP in a virtual machine can be specified by using the `ATTACH`, `LINK`, and `DEFINE` commands.
2. For more information about directory entries refer to the *VM/SP Planning Guide and Reference*.

Virtual Machine Configuration

To run the VM/SP nucleus in a virtual machine, load it onto the minidisk that represents the test system residence volume. Then, before initializing the system, verify that the virtual machine configuration has:

1. The correct console address
2. Sufficient unit record devices available at the correct addresses
3. Enough disks (either linked or attached) to make a reasonable test.

When setting up the virtual machine configuration, you can link to other user disks so that the real system can use these disks in its virtual operation. However, you must ensure that links to other disks use the same addresses and device types as were specified in the `DMKRIO` module of the test system.

For example, a real system has 3370s defined as 130 to 137 and has 3330s defined as 330 to 337. To avoid operational errors, the test VM/SP system links to user 3370 disks in the range of 130 to 137 and links to user 3330 disks in the range of 330 to 337. If your disk is linked at a 3370 address when it is actually a 3330 or 3340 device, the virtual VM/SP system issues errors when trying to process that disk. This happens because the address ranges correspond to the proper device type as described in the `DMKRIO` for Figure 19.

Defining a Console for VM/SP in a Virtual Machine

Since the logon console for a virtual machine operates as a 3215, 3210, 3270, or 1052, one of the following three methods can be used to satisfy the console requirements for your VM/SP virtual machine:

1. In the DMKRIO for the second level system you are building, define the console device as DEVTYPE 3215, 3210, or 1052 in the RDEVICE macro.

```
RDEVICE ADDRESS=01F,DEVTYPE=3215
```

2. Another approach is attaching a console-type device to your virtual machine and using that as your second level console. For example, if your DMKRIO for address 01F defines a DEVTYPE of 3277, then attach a real 3277 to your virtual machine as address 01F to function as your second level console.

3. Use the ALTCONS macro in your DMKRIO to specify an alternate 3270 console.

```
RIOGEN CONS=01F,ALTCONS=(009,010)
```

CMS System

For VM/SP in a virtual machine to also run CMS, it must have access to the CMS system residence volume. The virtual machine can access this volume either during logon (by using the LINK statement in the directory entry) or after logon (by using the CP LINK command). If passwords are provided, the test system can link to other users' disks so that they can be used by the CMS system. The one userid for the test system, can access all the disks necessary to do a VMFLOAD or any other similar function.

Accessing a VM/SP System Running in a Virtual Machine

Depending upon the nature of virtual machine testing, one or more graphic devices can be defined so that you may use the DIAL command to access the test system. In most cases, simple tests do not require any graphic devices to be defined or enabled at the virtual machine level. Most testing can be performed from the operator's virtual console, unless it is a 3215 which is a typewriter like device.

CP Disks for the Virtual Machine

Before VM/SP in a virtual machine can use the CP disks for the virtual system residence, paging, and spooling volumes, you must first format and allocate space for these disks.

Formatting the Volumes

To format the system residence, paging, and spooling volumes, use the CP Format/Allocate program. Although this program can run in a virtual machine, it cannot run under CMS. To run the format program, make it available to the virtual machine. Assuming you have made the stand-alone format utility available in your virtual reader, then IPL it from that reader (IPL 00C). Because a virtual disk is being formatted, the cylinder or block specification should reflect the size of the virtual disk being used. In the sample directory entry for TESTSYS, (Figure 2-1 on page 2-1) the MDISK statement for the virtual disk at address 330 defines only 50 cylinders for the device. Therefore, only 50 cylinders on the virtual disk at address 330 can be formatted.

When the second level system is going to use the same DMKSYS that the first level system is using, the virtual disk label should match the label in the CP-owned list. Thus, if you have two volumes in the owned list (such as CPDSK1 and CPDSK3), then those volume labels must match the minidisk labels used by the virtual machine. In our example in Figure 2-1 on page 2-1 we used MDISK 330/331. Also, DMKSYS must reflect the fact that the system residence volume could have a different layout. Make these changes to the SYSRES macro in DMKSYS.

Allocating Space for the Volumes

After formatting the volumes, allocate space on them for:

1. A directory on your test VM/SP system
2. Nucleus area
3. Warm start area
4. Error recording area
5. Paging space
6. Spooling space
7. Dump space

If the space is inaccessible to the test system (if it is beyond the size of the virtual disk), it must be assigned as permanent space. Assuming a 3330 Model 1, cylinders 51-403 on virtual device 330 (see Figure 2-1 on page 2-1) must be assigned as permanent space. This is necessary because the minidisk is smaller than the real device and by allocating it as permanent space, your test system will not access the area outside the minidisk. Otherwise, the virtual system attempts to access temporary space beyond the size of the virtual disk, resulting in the real system reflecting either seek checks or command rejects to your test system.

When allocating permanent space, organize the cylinders to hold the:

1. Directory
2. CP Nucleus
3. Error Recording Area
4. Warm Start Area.

Also, organize the areas to begin with the first cylinder or block available on the disk. If the real system residence volume (SYSRES) uses this same organization and volume label, then the disk you use for your test system residence volume can use the same DMKSYS and DMKRIO.

For example, if the real SYSRES does not match the SYSRES for the test system, you should tailor the test DMKSYS to your own needs. When operating in a virtual machine, it is preferable that the same installation modules be used. Using the same modules ensures that the testing environment matches the modules used in the real machine configuration. The only exception to this rule is the directory that appears on the virtual disk. The directory on the virtual disk cannot be the same as the real system directory because none of the labels and displacements for the user disks match.

To create the test directory for the test SYSRES volume, run CMS in the same virtual machine and have the test system user link to the CMS disk with the desired filename and with a filetype of DIRECT. The CMS DIRECT command will use this file to write the test system's directory out onto the test system's directory cylinders.

Note: The DIRECT file must contain sufficient directory entries to test VM/SP in a virtual machine environment.

Virtual IPL and Operation

If the DASD type of the real SYSRES volume is the same as the test system and the SYSRES packs have the same layout (for example, same values for SYSNUC, SYSWRM, etc.), you can copy the nucleus over using DDR. If not, you must use the VMFLOAD procedure (or any equivalent procedure such as GENERATE) to create an IPL'able nucleus in your reader to be written out onto your test system's SYSRES device. Make sure the DMKSYS text deck this procedure uses is the correct one describing the test SYSRES (as explained in the previous section).

You must verify that the virtual machine configuration matches (by using a QUERY VIRTUAL command), or is a subset of the DMKRIO defined for the system to be tested. Once this is done you can perform an IPL of the virtual disk containing the CP nucleus. In our example (Figure 2-1 on page 2-1), it is disk 330.

Note: Attention handling varies with the type of terminal used. Refer to the *VM/SP Terminal Reference* for a list and description of the terminals supported by VM/SP.

IPL VM/SP in the normal fashion, responding where required. Because the test system user cannot set the time-of-day clock, always reply "no" to the change time-of-day clock question. Under most circumstances, it is advisable to perform a cold start unless some specific function requiring a warm start is to be tested.

To place dumps of the test CP system into the test system's virtual reader you must:

1. Specify a test system's userid in the SYSDUMP operand of the SYSOPR system generation macro instruction in DMKSYS.
2. Initialize the test CP system by assuming the SET DUMP AUTO CP command (class B) by default.

Note: The test system's userid in the SYSDUMP operand should be OPERATOR, rather than the default of OPERATNS. OPERATOR helps you to readily identify your dumps. It also makes the dump immediately available to the OPERATOR virtual machine user for processing.

Once you IPL the virtual machine and log onto the operator's virtual machine, you can run other systems under this userid or enable graphic display. Enabling graphic display allows other users to dial into this system, log onto VM/SP in a virtual machine, and perform whatever actions they require.

Accessing Devices

Once you IPL the virtual machine, the devices not accessible to that machine at IPL time are considered offline. However, you can attach more devices to this machine and have them placed online, as required. For example, tape drives can be attached by the real machine operator to the virtual machine configuration at the address matching the configuration of the test CP system. You can easily change these virtual addresses to conform to your test system's DMKRIO by using the CP DEFINE command. The test VM/SP operator then issues the VARY ON CUU

command, and can ATTACH or DEDICATE (in the directory) the devices as needed. Use the same procedure for graphic devices, unit record equipment, or other devices.

Note: Most testing can be done by initializing and running tests from the operator's virtual machine without enabling any graphic devices. For full screen application such as XEDIT, FILELIST, etc., use one of two methods.

1. Define GRAF at a CUU defined as a 3270 (or other graphic device) and DIAL to the test system's virtual machine.
2. If the test system's console is using the ALTCONS = CUU (in DMKRIO) where CUU is a 3270 (or other graphic device), simply disconnect the OPERATOR and logon to another userid which gives you full screen capabilities. This is true provided you have defined your console as CUU prior to IPLing your test VM system.

Graphic and Spool Devices

Graphic devices and spool unit record devices can be created by using the CP DEFINE command. Before the test CP operator can attach these lines or devices to a virtual machine user logged on the second level system, they must first be placed online to the test CP system. Once online, they can be attached and used by virtual machines in the test CP system. Graphic lines can be attached directly to the test CP system for testing in that environment without using the CP DIAL command.

Virtual Disks

It is possible to use virtual disks in the test CP system; however, their setup is complex and requires careful coordination with the real directory of the real system. For example, if a virtual disk is moved and the real directory of the real system is changed but the virtual directory is not changed, **serious operating errors can occur**. Therefore, do not use production virtual disks in the test CP system unless they are required for a specific test.

Note: When a virtual machine is linked to virtual disks before the user IPLs a system to run in the virtual machine, the virtual disks appear to the test system as disks with a zero relocation factor. For CMS to access them at the virtual CP level, you must attach the disks at the CP level. Then the user can access them as though they were dedicated disks. Otherwise, accesses beyond this disk will cause the real CP system to present I/O errors in the form of seek checks or command rejects to the virtual CP system. This in turn, reflects the errors to the virtual operating system.

Spooling Considerations

If the virtual machine performs any spooling operations, the test CP system is also spooling (unless it has dedicated unit record devices). This double spooling operation is not a problem. The test CP detects that it is running in a virtual machine and at the end of each spooled output file issues a CP CLOSE command to the real CP. This produces real spooled output for virtual spool files.

Notice that double separators occur. For instance, the separator page on virtual printed output includes four pages. Two pages for the virtual CP system and two

more pages for the separator of the virtual machine on which the virtual CP system is running. The extra set of separator pages can be avoided by using the START command with the NOSEP option on the test system.

Because the virtual machine operation at this level is complex, there is no easy way of describing how to do all the functions. It requires careful study and analysis. At all times it requires an awareness of what level of virtual machine is operating and what function the user is trying to perform.

Example -- VM/SP Under VM/SP

The following sample terminal session illustrates how to run a test CP system in a virtual machine environment. The examples are commented to point out some of the more pertinent considerations.

```
logon testsys
ENTER PASSWORD:

LOGMSG - 17:20:14 EDT THURSDAY mm/dd/yy
*****
* OPERATOR: MR. NICE..... SYSTEM STATUS PHONE: x9999 *
* FOR PROBLEM ASSISTANCE ..... x8888 *
* PLEASE DETACH TAPES WHEN FINISHED AND PURGE ALL *
* UNNECESSARY FILES TO AVOID SPOOL PROBLEMS. *
*****
FILES: NO RDR, NO PRT, NO PUN
LOGON AT 17:38:06 EDT THURSDAY mm/dd/yy
VM/SP 3.0 CMS
```

Figure 2-2. VM under VM: Logon procedure

Figure 2-2 shows a normal logon procedure for a user identified as TESTSYS. This userid is defined in the real CP directory with sufficient options and resources to run VM/SP in a virtual machine environment.

```
query virtual
STORAGE = 04096K
CHANNELS = BMX
RDR 00C CL A NOCONT HOLD EOF          READY
PUN 00D CL A NOCONT NOHOLD COPY 001   READY FOR STANDARD
    00D FOR TESTSYS DIST 999/999/
PRT 00E CL A NOCONT NOHOLD COPY 001   READY FOR STANDARD
    00E TO SYSTEM DIST 999/999/ FLASHC 000
    00E FLASH CHAR MDFY 0 FCB
CONS 01F ON GRAF 047 TERM START
    01F CL 1 NOCONT NOHOLD COPY 001   READY FOR STANDARD
    01F TO TESTSYS DIST 999/999/ FLASHC 000
    01F FLASH CHAR          MDFY      0 FCB
DASD 190 3375 CMS190 R/O          070 CYL
DASD 19D 3375 CMS190 R/O          042 CYL
DASD 19E 3375 CMS190 R/O          065 CYL
DASD 330 3330 PIDSK4 R/W          050 CYL
DASD 331 3330 PIDSK4 R/W          050 CYL
R;
```

Figure 2-3. VM under VM: Verifying the Virtual Machine Configuration

After logon, issue the QUERY VIRTUAL command to verify the virtual machine configuration. The response indicates:

- The storage size is 04096K bytes.
- Some unit record devices have been defined.
- The console is 01F and is real device 047.
- Devices 190, 19D and 19E are available to operate CMS in this virtual machine.
- Devices 330/331 are the 3330, 50-cylinder, read/write minidisks that become the test system residence volumes for this virtual machine when it is running VM/SP.
- The volume serial numbers (volids) for the DASD units are those of the real disks on the real computing system.

```
link usecms 191 191 rr
ENTER READ PASSWORD:

DASD 191 LINKED R/O; R/W BY USECMS
R;
```

Figure 2-4. VM under VM: Accessing a virtual machine's minidisk

The LINK command in Figure 2-4 allows you to access the USECMS virtual machine's minidisk. This minidisk is the CMS disk containing certain directory files.

```
def 01f as 009
CONS 009 DEFINED

i cms
VM/SP 3.0 CMS...FLOOR...mm/dd/yy

Y (19E) R/O
A (191) R/O
R;
```

Figure 2-5. VM under VM: Redefining the Console

Figure 2-5 shows the redefining of console 01F as 009 before issuing the CP IPL command.

```
listf * direct a
TESTSYS DIRECT A1
USERTEST DIRECT A2
USER DIRECT A1
USER1 DIRECT A1
R;
```

The LISTFILE command, issued in the CMS environment, shows that there are four files with a filetype of DIRECT. For our discussion we need to XEDIT file TESTSYS DIRECT, (as shown in the following example).

```
TESTSYS  DIRECT  A1  F 80  TRUNC=72  SIZE=20  LINE=9  COLUMN

* * * TOP OF FILE * * *
===== DIRECTORY 330 3330 VMSRES
===== USER OPERATOR OPERATOR 1M 2M ABCDEFG
===== ACCOUNT 12345678 COMP.RM
=====  CONSOLE 9 3215
=====  SPOOL C 2540 READER A
=====  SPOOL D 2540 PUNCH A
=====  SPOOL E 1403 A
=====  LINK MAINT 190 190 RR
=====  MDISK 191 3330 10 9 USECMS RR RDGDEV WDGDEV MDGDEV
|...+...1...+...2...+...3...+...4...+...5...
===== USER MAINT CPCMS 1M 16M ABCDEFG
===== ACCOUNT 12345679 ROOM331
=====  CONSOLE 9 3215
=====  SPOOL C 2540 READER A
=====  SPOOL D 2540 PUNCH A
=====  SPOOL E 1403 A
=====  MDISK 191 3330 10 9 USECMS WR RDGDEV WDGDEV MDGDEV
=====  MDISK 196 3330 0 10 SYS196 RR RDGDEV
=====  MDISK 190 3375 0 70 CMS190 RR RDGDEV
=====  MDISK 19E 3375 0 65 FLRCMS RR RDGDEV
=====  DED 19A CMS19A
====>
```

Notice the DIRECTORY statement. It specifies that the directory is to be written on a 3330 device at address 330 with disk label VMSRES. This address corresponds to the 3330 minidisk that was shown and discussed in Figure 17. Because this is a minidisk, VMSRES is the label for that minidisk not a label on a complete real disk. Therefore, the virtual 3330 disk (VMSRES) is on the real disk labeled PIDSK4. This minidisk was previously formatted, labeled, and allocated by the CP Format/Allocate program. For information on the Format/Allocate program see the *VM/SP CMS User's Guide*.

Note: The user identified as OPERATOR has all privilege classes to control the test system. The console and unit record devices are defined to allow him to operate CMS. The minidisks defined for this userid have a displacement of zero and a size that does not exceed the bounds of the minidisks defined for the test system. The volids specified on the directory statements are the volids on the virtual disks for the test CP system. They are not the volids of the real disks on which those virtual disks are defined for the test CP system.

```
direct testsys
EOJ DIRECTORY UPDATED
R(00006);
```

The above example shows the operation of the directory program in a virtual machine. The file used to create the test directory is TESTSYS DIRECT. Notice that the return code is 6. The directory has been updated on the disk, but because

this disk is a virtual disk and not the real system residence disk, the real CP system directory has not been modified. The return code of 6 is the normal return code indicator.

```
det 191
DASD 191 DETACHED
R;
```

Since the 191 disk of user USECMS is no longer needed, it is now detached.

```
link cpsys 196 196 rr
ENTER READ PASSWORD:

R;
link cpsys 194 194 rr
ENTER READ PASSWORD:

R;
acc 196 a
'196' REPLACES ' A (191) '.
A (196) R/O.
R;
acc 194 b/a
B (194) R/O.
R;
```

You must now do a VMFLOAD function, but first you must access the disks needed to do a VMFLOAD. The LINK commands define the disks that contain the necessary CMS files to build a CP system to be tested in a virtual machine environment. The CMS ACCESS command places those disks in a read-only status. In the above example IMSG has been set off; therefore, none of the information messages appear.

```
spool pun *
R;
```

The CP SPOOL command transfers the output of the spool punch back to this userid. This is required so that you may later IPL the virtual card reader to load the CP nucleus onto the test system residence disk.

```
spool prt *
R;
```

The CP SPOOL command transfers the output of the printer back to this userid. This transfer is required so that the user may later read in the nucleus load map.

```
vmfload cpload dmksp  
SYSTEM LOAD DECK COMPLETE  
PUN FILE 0189 TO TESTSYS COPY 001   NOHOLD  
R;
```

The VMFLOAD function is run specifying the load list CPLOAD and a control file of DMKSP. DMKSP is a special control file used to apply experimental updates and PTFs. At the completion of the load function, the spool file is transferred to TESTSYS and is available as a reader file.

```
ipl 00c  
NUCLEUS LOADED ON VMSRES --- STARTING CYL/BLK=032   , LAST CYL/BLK USED= 037  
CP ENTERED; DISABLED WAIT PSW '00020000 00000012'
```

The NUCLEUS LOADED ON VMSRES message indicates that the nucleus has been loaded onto the virtual disk. You are now finished using CMS for the directory and IPL deck setup. The IPL of card reader 00C loads the nucleus, and the loader is distributed with the following default I/O addresses:

```
CONSOLE = 009  
PRINTER = 00E
```

```
close prt  
PRT FILE 0190 TO TESTSYS COPY 01   NOHOLD
```

In the above example, the CLOSE command causes VM/SP to place the nucleus load map in the virtual reader. The minidisk label must either be VMSRES or be defined in DMKSYS. The virtual machine enters the disabled wait state after producing a message from the real CP system.

```
def 009 as 01f  
CONS 01F DEFINED  
R;
```

The console must be redefined as 01F. This is the console address that was specified in DMKRIO, and was loaded as part of the CP nucleus.

i cms

VM/SP 3.0 CMS...FLOOR...mm/dd/yy

Y (19E) R/O

R;

receive 0198 testnuc map a

File TESTNUC MAP A1 received from TESTSYS at NODEID sent as NONE NONE A

R;

Initializing CMS and receiving TESTNUC MAP places the nucleus load map on the test userid's (TESTSYS) CMS A-disk.

query virtual

STORAGE = 04096K

CHANNELS = BMX

RDR 00C CL A NOCONT HOLD EOF READY

PUN 00D CL A NOCONT NOHOLD COPY 001 READY FOR STANDARD
00D FOR TESTSYS DIST 999/999/

PRT 00E CL A NOCONT NOHOLD COPY 001 READY FOR STANDARD
00E TO SYSTEM DIST 999/999/ FLASHC 00D
00E FLASH CHAR MDFY 0 FCB

CONS 01F ON GRAF 047 TERM START
01F CL 1 NOCONT NOHOLD COPY 001 READY FOR STANDARD
01F TO TESTSYS DIST 999/999/ FLASHC 000
01F FLASH CHAR MDFY 0 FCB

DASD 190 3375 CMS190 R/O 070 CYL

DASD 191 3375 PIDSK3 R/W 010 CYL

DASD 194 3330 PIDSK5 R/O 060 CYL

DASD 196 3330 PIDSK7 R/O 010 CYL

DASD 19D 3375 CMS190 R/O 042 CYL

DASD 19E 3375 CMS190 R/O 065 CYL

DASD 330 3330 PIDSK4 R/W 050 CYL

R;

The QUERY VIRTUAL command displays the current virtual machine configuration. This is the configuration that was used to run the CMS machine, except that the console address has been changed to 01F. Before initializing the virtual 330 disk and IPLing VM/SP, it is necessary to redefine the disk addresses so that they can be recognized by the test system.

```

define 190 as 130
DASD 130 DEFINED
R;
define 194 as 331
DASD 331 DEFINED
R;
define 196 as 332
DASD 332 DEFINED
R;
define 19e as 131
DASD 131 DEFINED
R;
link virttest 191 333 r
ENTER READ PASSWORD:

DASD 333 LINKED R/O
R;

```

These DEFINE commands and the LINK command change the configuration of the virtual machine so that it can be recognized by the test nucleus. Notice that the 3375s are defined in the range of addresses 130 to 137 and that the 3330s are defined in the range of 330 to 337. The LINK command is used to access another user's disk as a 3330 at address 333.

```

query virtual
STORAGE = 04096K
CHANNELS = BMX
RDR 00C CL A NOCONT HOLD EOF          READY
PUN 00D CL A NOCONT NOHOLD COPY 001    READY FOR STANDARD
    00D FOR TESTSYS DIST 999/999/
PRT 00E CL A NOCONT NOHOLD COPY 001    READY FOR STANDARD
    00E SYSTEM DIST 999/999/ FLASHC 000
    00E FLASH CHAR MDFY 0 FCB
CONS 01F ON GRAF 051 TERM START
    01F CL 1 NOCONT NOHOLD COPY 001     READY FOR STANDARD
    01F TO TESTSYS DIST 999/999/ FLASHC 000
    01F FLASH CHAR MDFY 0 FCB
DASD 130 3375 CMS370 R/O      056 CYL
DASD 131 3375 CMS190 R/O     026 CYL
DASD 19A 3375 CMS190 R/O     055 CYL
DASD 290 3375 PIDSK3 R/O     045 CYL
DASD 330 3330 PIDSK4 R/W     020 CYL
DASD 331 3330 PIDSK5 R/O     060 CYL
DASD 332 3330 PIDSK7 R/O     010 CYL
DASD 333 3330 PIDSK7 R/O     010 CYL
R;

```

A CP QUERY VIRTUAL command is issued again to show that the virtual machine configuration has been redefined to match one that can be recognized by the test system. Notice that the 330 disk has read/write status (this is required for VM/SP to do paging and spooling). All the other disks have read-only status. Disks 19A and 290 are not recognized by the test system because they are not defined in its DMKRIO; however, their inclusion in the configuration does not matter.

ipl 330

The test system is loaded by an IPL of the test system residence volume (330).

```
VM/SP RELEASE 3, SERVICE LEVEL 0000; 06/16/83 09:19:39
NOW 09:21:18 EDT THURSDAY 06/16/83
CHANGE TOD CLOCK (YES|NO) : no

DMKCPI971I SYSTEM IS UP GENERATED
09:21:26 START ((COLD|WARM|CKPT|FORCE) (DRAIN))|(SHUTDOWN) : cold
09:21:26

DMKCPI953I UNABLE TO ALLOCATE SYSTEM AUTO DUMP
09:21:26 DMKLNK108E MAINT 19E NOT LINKED; VOLID FLRCMS NOT MOUNTED
RRRR....RING....GGGG
09:21:26 AUTO LOG *** OPERATOR USERS = 001 BY SYSTEM
09:21:26
RRRR....RING....GGGG

DMKCPI951I CP VOLID VMSEXT NOT MOUNTED
09:21:26
RRRR....RING....GGGG

DMKCPI951I CP VOLID VMPK01 NOT MOUNTED
09:21:26
RRRR....RING....GGGG

DMKCPI957I STOR 04096, NUC 348K, DYN 03436K, TRA 060K, FREE 0252K, V=R 00000K
09:21:26 FILES: NO RDR, NO PRT, NO PUN
RRRR....RING....GGGG
09:21:30 AUTO LOGON *** AUTOLOG1 USERS = 002 BY OPERATOR

DMKCPJ966I INITIALIZATION COMPLETE
```

This output is from the VM/SP system running in a virtual machine. It is printing the responses on what appears to it as a virtual 3215 console. Notice that the prompting CHANGE TOD CLOCK (YES/NO) does not require a response. If you were to respond with a yes, it would request a date and time to be set; however, the real time-of-day clock cannot be changed from a virtual machine environment. The LINK error messages are a result of the automatic operator logon and of the directory not being able to find some disks defined in the operator's virtual machine. The "RING" message is the real CP simulation of the virtual console alarm function. Finally, the operator receives confirmation of logon.

VM/SP issues the messages indicating that CPDRM1 and PIDSK2 are not mounted because the test DMKSYS has an owned list (SYSOWN macro in

DMKSYS) that has three volumes specified: CPDRM1, PIDSK2, and PIDSK3. The only one available in the configuration during IPL was the system residence volume VMSRES. These error messages are not severe; only a minimum amount of space is required by CP to accomplish paging and spooling. The response to the start message in this case is “cold”. This is the normal response unless a specific test of warm start is required.

```

query dasd all
09:21:36 DASD 130 CP SYSTEM CMS190      001
09:21:36 DASD 131 FREE
09:21:36 DASD 132 OFFLINE
09:21:36 DASD 133 OFFLINE
09:21:36 DASD 134 OFFLINE
09:21:36 DASD 135 OFFLINE
09:21:37 DASD 136 OFFLINE
09:21:37 DASD 137 OFFLINE
09:21:37 DASD 250 OFFLINE
09:21:37 DASD 251 OFFLINE
09:21:37 DASD 252 OFFLINE
09:21:37 DASD 253 OFFLINE
09:21:37 DASD 254 OFFLINE
09:21:37 DASD 255 OFFLINE
09:21:37 DASD 256 OFFLINE
09:21:37 DASD 257 OFFLINE
09:21:37 DASD 2D0 OFFLINE
09:21:37 DASD 2D1 OFFLINE
09:21:37 DASD 2D2 OFFLINE
09:21:37 DASD 330 CP OWNED  PIDSK4      001
09:21:38 DASD 331 CP SYSTEM CPRL10     001
09:21:38 DASD 332 CP SYSTEM SYS196     001
09:21:38 DASD 333 FREE
09:21:38 DASD 334 OFFLINE
09:21:38 DASD 335 OFFLINE
09:21:38 DASD 336 OFFLINE
09:21:38 DASD 337 OFFLINE
09:21:38 DASD 350 OFFLINE
09:21:38 DASD 351 OFFLINE
09:21:38 DASD 352 OFFLINE
09:21:38 DASD 353 OFFLINE
R;

```

The CP test system issues a read. The response to the read is the entry of the QUERY DASD command. The test CP system responds with the status, as shown in the figure above. Notice that most of the devices are in an offline condition, since at the time of the IPL these device addresses were not available in the virtual machine configuration. The devices that were available are now marked free, owned, or system. (The system volumes are ones that have minidisks in use by the operator.) Notice that device 332 has a label of SYS196 in the test CP system. A previous QUERY VIRTUAL showed that DASD 332 is actually physically mounted on PIDSK7. However, this label is the real system label and is not the one recognized by the test CP system. For users to access the 332 disk, the test directory must refer to the virtual label of SYS196. (MAINT’s minidisk 196 refers to a zero cylinder displacement on volume SYS196.)

q virtual

```
09:21:46
09:21:47 STORAGE = 01024K
09:21:47 CHANNELS = SEL
09:21:47 CONS 009 ON CONS 01F TERM STOP
09:21:47 009 CL T NOCONT NOHOLD COPY 001 READY FOR STANDARD
09:21:47 009 FOR OPERATOR DIST OPERATOR FLASHC 000
09:21:47 009 FLASH CHAR MDFY 0 FCB
09:21:47 RDR 00C CLS * NOCONT HOLD EOF READY
09:21:47 PUN 00D CLS A NOCONT NOHOLD COPY 001 READY FOR STANDARD
09:21:47 00D FOR OPERATOR DIST OPERATOR
09:21:48 PRT 00E CLS A NOCONT COPY 001 READY FOR STANDARD
09:21:48 00E FOR OPERATOR DIST OPERATOR FLASHC 000
09:21:48 00E FLASH CHAR MDFY 0 FCB
09:21:49 DASD 190 3375 CMS190 R/O 056 CYL
09:21:49 DASD 191 3375 PIDS3 R/W 010 CYL
09:21:49 DASD 196 3330 SYS196 R/O 010 CYL
09:21:49 DASD 196 3330 SYS196 R/O 010 CYL
R;
```

Signal attention to the test CP system. The operator types in **QUERY VIRTUAL**, and the display is the virtual machine configuration for the virtual machine operator. Note that the operator has a configuration that is suitable for running CMS by loading (via IPL) virtual device 190.

att 131 operator 191

```
DASD 131 ATTACH TO OPERATOR 191
R;
```

The operator attaches to himself what appears to him as a real 131 as virtual address 191. The following response indicates a successful attach.

cp q v 131

```
DASD 131 3375 CMS190 R/O 065 CYL
R;
```

The response to your query virtual indicates that the virtual 131 disk is a 3375 with read-only status, and has 65 usable cylinders.

q 131

```
DASD 131 ATTACH TO OPERATOR 191
```

q v 191

```
DASD 191 ON DEV 131
```

Signalling attention takes you back to the virtual machine level, where an attention interrupt is reflected. The test CP system then responds with a read. At this level you must issue a **QUERY 131**. For the operator it is a query of what appears to him as real disk 131. Note that the status is that of the disk attached to the

operator as virtual address 191. This is the same disk that was previously noted; however, the test CP system thinks that the disk has read/write status. Signal attention again to cause a read, now you can issue a QUERY VIRTUAL 191. The response indicates a dedicated disk on device 131 and assumed read/write status.

```
ipl 190  
VM/SP 3.0 CMS...FLOOR...mm/dd/yy  
  
DMSACC112S 'A (191) ' DEVICE ERROR  
R;
```

Signalling attention causes a CP read; the operator must perform an IPL of the virtual 190 disk to load the CMS system. The response is from the CMS system that is running in a virtual machine under the test system running under a real VM/SP system. A null response to the ensuing read gives an error message from CMS. The error message appears because CMS has an indication from the test CP system that it has write access to the disk (remember it appears as a dedicated disk). However, the real CP system has the disk in read-only status and rejects the write attempt to the test CP system. This in turn reflects it to CMS, causing the device error message.

```
#cp det 333  
DASD 333 DETACHED  
R;  
#cp link virtest 191 333 w  
ENTER WRITE PASSWORD:  
  
DASD 333 LINKED R/W  
R;
```

Figure 2-6. Linking 333 from read only to write

Enter the real CP mode by signalling attention. Detach device 333 and link to it as 333 in write mode. The fact that the operator detached and relinked is transparent to the test CP system at this level. You have accomplished a status change from read to write. The physical extent definition has not changed.

```
det 191  
#cp att 333 operator 191  
DASD 131 DETACHED OPERATOR 191  
DASD 191 DETACHED  
DASD 333 ATTACH TO OPERATOR 191  
  
b  
CMS  
  
acc 191 a  
R;
```

The operator detaches the virtual 191 disk and attaches the real 333 disk to his userid as 191. Note that the 333 appears to the test CP system as a real disk, when it actually is a virtual disk. The BEGIN command (b) changes the virtual machine environment to CMS. The ACCESS 191 command is then successfully completed, giving write access to the virtual 191 disk, which is the test CP system's 333 disk previously linked in write mode.

print profile exec

```
PRT 00E OUTPUT OF OPERATOR FILE=0002 LINES=00013  
R;
```

drain 00e

```
PRT 00E SPOOL CLS XA DRAINED
```

From CMS, the PROFILE EXEC is printed. The test CP system responds with a printer output message for file 2, which is the output from the previous print function. The ready message is the response from the CMS system. The above example shows a virtual machine running with a virtual console that is receiving both virtual machine and CP messages. Signaling attention places the virtual machine into test CP mode, where you can specify a drain of device 00E. The system responds with a message indicating that the device is drained. This indicates that the test CP system has completed printing on what it thinks is a real printer. This printer is actually spooled by the real CP system.

set dump auto CP

q dump

```
DASD 330 DUMP UNIT CP TO TEMP
```

Signalling attention takes you to the test CP system level, where you issue a SET DUMP command. Ordinarily, when testing an unstable system, this would have been one of the first commands entered after issuing the IPL for the test CP system. The query of the dump unit verifies that the dump is of the CP nucleus to the spooling disk at address 330.

ipl 190

09:21:59

VM/SP SL202 '8301' CMS 06/16/83

CMSSEG SYSTEM NAME 'CMSSEG' NOT AVAILABLE

R; T=0.01/0.01 09:22:22

set dump auto cp

09:22:29 UNABLE TO ALLOCATE SYSTEM AUTO DUMP

RRRR....RING....GGGG

R900953); T=0.01/0.02 09:22:29

CP SYSTEM RESTART

RRRR....RING....GGGG

09:22:41 DMKDMP908I SYSTEM FAILURE; CODE PSA002 PROCESSOR 00

RRRR....RING....GGGG

RRRR....RING....GGGG

DMKCKP960I SYSTEM WARM START DATA SAVED

DMKCKP961W SYSTEM SHUTDOWN COMPLETE

RRRR....RING....GGGG

CP ENTERED; DISABLED WAIT PSW '000A0000 00000008'

Signalling attention takes you to the real CP level, where you enter the **SYSTEM RESTART** command. This command is the equivalent of a system restart function on a real processor. The system restart function for a CP system automatically dumps the system and then issues an automatic IPL. After the system is dumped, a message appears with abnormal termination code PSA002 (a system dump due to pressing the system restart key).

The virtual bell rings to indicate that the system has been reloaded, and the system prints messages about: saving warm start data, CP entering a disabled wait state, and system shutdown being complete. The message indicating that CP has entered a disabled wait state is prematurely issued between these two messages. It occurs because of a synchronization of the real CP system with the test CP system console output.

After these messages are issued, you are in real CP mode. You can either log off or obtain the system abend dump.

To obtain the system abend dump, re-IPL 330 and repeat the test procedure up to the point where the 'print profile exec' is shown in the same session. At this point, you now have CMS initialized in the virtual CP system and have read/write access to your real CMS minidisk at virtual address 191. By issuing a **QUERY RDR ALL** command, VM/SP should reveal that a class D dump file is in the operator's virtual reader (because the operator's userid is specified in the **SYSDUMP** operand of the **SYSOPR** system generation macro instruction).


```
spool rdr cl d
b
CMS
ipcsdump
.
.
.
R;
```

By entering the IPCSDUMP command (class C or E), IPCS reads the CP abend dump and creates a CMS symptom record and dump file, problem report, and symptom summary entry on the 191 A-disk. For a sample IPCSDUMP session, refer to *VM/SP IPCS User's Guide*. When IPCSDUMP completes its processing under CMS in the test CP system, terminate the test system by entering real CP mode and initializing CMS. Once under CMS, you can issue the IPCS DUMPSCAN command to look at the dump taken of your test system. This dump resides as a dump file on your real 191 A-disk (USERID VIRTEST, from the write LINK in Figure 2-6 on page 2-17).

Summary

To update and test a VM/SP system in a virtual machine, you must first have a VM/SP directory entry for a test VM/SP virtual machine. A test system directory (usually a separate and small subset of the real directory) must also exist. The test system directory need only specify the minimum number of users to perform the test. Before initializing this system, an installation should verify that the virtual machine configuration has:

- The correct console address
- Sufficient unit record devices available at the correct addresses
- Enough disks (either linked or attached) to make a reasonable test to run CMS when it has access to the CMS system residence volume.

With few exceptions, the IPL for a VM/SP virtual machine is similar to IPLing a real VM/SP system. Operationally, VM/SP provides CP commands to display and store into real storage. The VM/SP system in a virtual machine can also display and store into its own third level virtual storage. If the virtual machine performs any spooling operations, the test VM/SP system is also spooling unless it has dedicated unit record devices. This double spooling is no problem.

Section 3. DOS in a Virtual Machine

When loading DOS into a virtual machine running under VM/SP, the VM terminal becomes the DOS operator console, and the user is responsible for entering all the commands and responses normally required of the operator.

The four basic techniques to use when running DOS in a virtual machine are:

- You must establish in the CP directory two userid's; one as the primary userid and the other as the secondary userid. Logon to your primary userid and IPL DOS. Once it is running, disconnect from your primary userid and logon to the secondary userid. Now both virtual machines can be operated from one terminal.
- Run DOS in batch mode. The terminal is the operator console and other users may submit jobs either through the real system card reader(s) or from the virtual card punches of other userids.
- Use the IPL command to alternate between using DOS and CMS in a single virtual machine. Use CMS to prepare a job stream for DOS, use DOS to execute the job stream, and use CMS to check the output.
- Logon to a userid and load DOS. Once it is running, disconnect from the DOS userid and logon to a second userid while the DOS userid continues working. To check on the status of DOS, disconnect from the current virtual machine and reconnect the DOS virtual machine.

Before discussing the above techniques in greater detail, you must understand how to:

- Create VM/SP directory entries for DOS virtual machines
- Access the DOS system residence volume
- Ensure that the proper I/O devices are attached to the DOS virtual machine
- IPL and operate DOS under VM/SP.

Note: Multiple DOS statements cannot be entered on a single line using the logical line end character (#). All logical line end characters translate to a X'15' before being passed to a virtual machine; DOS does not recognize this condition.

System Generation Recommendations

When generating DOS/VS to run in a virtual machine, you should have these primary objectives:

- To reduce the number of start I/O instructions (SIO) issued by DOS, including those issued for DOS paging I/O operations.
- To avoid double CCW translation you need to consider how to generate both VM/SP and DOS.

Note: The following recommendations have been made by users who run DOS in a virtual machine under VM/SP. As such, these recommendations have not been submitted to any formal test by IBM. Prior to any implementation, you should evaluate their usefulness in your own situation.

VM/SP Recommendations

When generating VM/SP for a DOS virtual machine, note the following recommendations:

VM/SP Saved Systems

IPL time can be reduced by saving any operating system after the generated operating system has been loaded on VM/SP. For more information about generating saved systems, refer to the *VM/SP System Programmer's Guide*.

Handshaking for DOS

DOS Release 34 with the Advanced Functions-DOS Program Product (Program No. 5746-XE2) uses VM/VS handshaking. For further details, refer to the appropriate DOS program product publications. VSE with the VSE/Advanced Functions Program Product (5746-XE8) uses VM/VS handshaking (also known as the VSE-VM/370 linkage facility). For further details, refer to *VSE/Advanced Functions General Information*, GC33-6106.

Initializing Disks and Minidisks

To initialize a DOS minidisk for use under VM/SP, the Device Support Facilities service program *must* be used. This stand-alone utility is shipped with VM/SP as CMS file 'IPL DSF S2'.

DOS Recommendations

When generating DOS to run in a virtual machine, note the following recommendations:

Generating a DOS Supervisor

- **Specifying boundary size (DOS/VS Release 34 and earlier only)**

When generating a DOS supervisor, use 4K (the size of VM/SP's pages) whenever DOS recommends using a 2K boundary or a multiple of 2K. Do not use the default for the SEND macro instruction. It causes DOS to round the supervisor size to the next 2K boundary. Instead, manually calculate the size of the supervisor and specify a 4K boundary in the SEND macro instruction. This specification forces DOS to be loaded at the next 4K page boundary.

- **Reducing privileged instructions**

Improve performance by reducing the number of privileged instructions that must be handled by VM/SP and virtual machine assist. Generate a tailored DOS supervisor for each virtual machine and leave out any unnecessary options. Because VM/SP issues its own stand-alone seek (except for 2314 disks), do not specify seek separation in the FOPT macro instruction.

When using DOS/VS Release 34 (or earlier) and your processor has block multiplexer channels, specify the block multiplexer option in both the PIOCS macro instruction and the VM/SP directory OPTION statement. However, this PIOCS specification in PIOCS is unnecessary when using the VSE environment because block multiplexer support is standard.

Specifying DOS Partition Sizes (In DOS Systems without VM/VS Handshaking)

If you are running a DOS system that does not have VM/VS handshaking, sometimes performance is better when using several virtual machines rather than using many active partitions in one virtual machine. For example, if your installation has a communications system, a batch system, and a test system, create a separate virtual machine for each one. However, to only run VM/SP part of the day and to minimize operational differences, one multipartition production DOS machine may be preferable.

It is usually best to make the DOS partition sizes, and thus the whole DOS virtual machine, large enough so that all jobs run V=R. Let VM/SP do the paging.

Rotational position sensing (RPS) cannot be used with V=R partitions. Also, avoid double CCW translation and double paging.

Set RSIZE equal to the supervisor size plus the sum of all V=R partitions, plus the SVA, plus 32K.

Note: You must specify, "REAL" on the DOS // EXEC job control card.

Generating the Operating System

When VM/SP is the primary operating system and DOS is running one or two partitions in a virtual machine, generate DOS with as few options as possible. This is particularly true when several virtual machines share the same system residence volume.

When VM/SP is not the primary operating system and DOS is running without VM/SP, generate DOS to:

- Be transparent to the users of the other systems
- Have the required number of partitions

Generating POWER

When DOS is run under VM/SP, POWER (POWER/VS for DOS or VSE/POWER for the VSE environment) should be used with the appropriate unit record devices that are dedicated to DOS.

If an installation has sufficient DASD space, let both POWER and VM/SP spool. Generate POWER with only the options that suit the installation's needs; but make the I/O buffer sizes as large as possible, up to 2008 bytes. If one job step in a DOS job stream abends, it is easy to use POWER to cancel the remainder of the job stream. To use only VM/SP spooling, an installation must manually cancel each job step.

Sharing the DOS System Residence Volume

The recommended method of sharing a DOS System Residence DASD volume (SYSRES) is to have each virtual machine IPLing DOS, have a separate minimal DOS SYSRES accessed R/W, and then have R/O access to all private DOS libraries.

Reducing SLD Directory Reading

In the FOPT macro instruction, specify enough second level directory (SLD) entries to reduce repetitious reading of the directory.

Use the system directory list (SDL) in the shared virtual area (SVA) for all job control, disk and tape open and close transients, and for the attention routine.

Executing Programs Under DOS and CMS/DOS

If a program in a DOS core image library may be executed in either a DOS virtual machine or under CMS/DOS, link edit the program with the ACTION REL linkage editor control statement so that it uses the DOS relocating loader.

DOS Accounting

Note: This topic applies only when running DOS Release 34 (or earlier). It does *not* apply when running either DOS Release 34 with the Advanced Functions-DOS Program Product (5746-XE2) or running in the VSE environment.

Except with processors that have ECPS: VM/370 and virtual interval timer assist, DOS accounting gives inaccurate and inconsistent elapsed processor times when operating under VM/SP with virtual machine assist. This inconsistency occurs because the interval timer (located at virtual storage location X'50') used by the DOS accounting routine is only updated when VM/SP gets control. Therefore, when DOS accesses the interval timer data, a variable amount of time may have elapsed since VM/SP last updated the interval timer, and thus DOS records an inaccurate processor time.

To attempt to minimize this inaccuracy at the cost of some additional VM/SP overhead, you may wish to add the following dummy DIAGNOSE instruction:

83000000

at the following locations in the DOS supervisor source statements:

- In the SVC 24 routine, before the L R3,SYSTIMER statement
- In the SVC 52 routine, before the L R3,SYSTIMER statement
- In the STCLOCK routine, before the STCK CLOCK statement
- In the timer interruption handler routine, before the LM R2,R3,SYSTIMER statement
- In the job accounting initialization routine (JATIMER), before the statement that references SYSTIMER

Note: To prevent a possible specification exception to DOS, ensure that general register zero contains zeros before issuing the DIAGNOSE instruction.

If VM/SP is running on a processor model that has ECPS: VM/370 (as defined in the *VM/SP Planning Guide and Reference*), enable virtual interval timer assist. This action lets the hardware, rather than VM/SP, update the virtual interval timer. Hardware update frequency is 300 times per second and results in accurate and repeatable time measurements.

Sample DOS Directory Entries

The following directory entries represent some batch type virtual machines that can be used to run production jobs under DOS. The operands specified on the OPTION control statement reflect the requirements of the particular system being used. Disk space can either be dedicated or shared with other systems.

The following directory is for a VSE/AF guest machine.

```
USER VSEAF U 6M 8M G
ACCOUNT 9999 DKH/15D/
IPL 250
OPTION REALTIMER ECMODE BMX
CONSOLE 01F 3215 T VSEMAINT
SPOOL 00C 2540 READER A
SPOOL 00C 2540 PUNCH A
SPOOL 00E 1403 A
MDISK 250 3330 000 404 VSERES WR ALL DOSWRITE
MDISK 251 3330 404 404 VSERES WR ALL DOSWRITE
```

The following directory entry is for secondary users to control VSE/AF when it is running disconnected.

```
USER VSEMAINT U 1536K 16M G
ACCOUNT 9999 DKH/1501
IPL CMS
CONSOLE 009 3215
SPOOL 00C 2540 READER A
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 PRINTER A
LINK MAINT 190 190 RR
LINK MAINT 19D 19D RR
LINK MAINT 19E 19E RR
LINK VSEAF 250 250 RR
LINK VSEAF 251 251 RR
MDISK 191 3375 001 050 VSEU01 WR ALL DKHWRITE
```

Accessing DOS

This topic assumes that DOS for use under VM/SP has already been generated and that the system residence volume is available on a real disk or minidisk in read/write status. A user can make the system residence volume available in any one of these ways:

- Defining the DOS system residence as a read/write disk in the VM/SP directory entry for the userid running DOS. A typical directory entry might look like the following:

```
MDISK 250 3330 101 50 VDOSYS MR RPASS WPASS
```

- Linking to the DOS system residence volume using the LINK command. For example, if the DOS system residence is on the 150 disk in the directory entry for the userid DOSRES, you could enter:

```
link dosres 150 250 w wpass
```

- Having the VM/SP system operator attach the DOS system residence directly to a userid, for that user's exclusive use. When the operator (or another user with Class B command privileges) attaches the disk (not a minidisk) to one user, no one else may access the volume.

Note: All of the console logs and command examples in this section assume that a DOS system residence is attached to the virtual machine at virtual address 250.

Using Virtual Unit Record Devices

When using DOS in a virtual machine, a user must have the following unit record devices, which are normally defined in the VM/SP directory entry:

- A virtual card reader, from which DOS reads the job input stream.
- A virtual printer, that receives all the SYSLST output generated during DOS operation.
- A virtual card punch, that receives SYSPCH output generated during DOS operation.

Depending upon how DOS was generated, a user may need to determine a virtual device address. For example, if DOS expects a 3211 printer at address 002 and no printer is at this address in the virtual machine configuration, define one with the CP DEFINE command:

```
define 3211 002
```

Note: When using CMS to prepare jobs for a DOS virtual machine, use your virtual card punch to spool jobs to the DOS virtual machine.

Before using DOS, find out (from the programmer at the installation responsible for generating and maintaining DOS) what the virtual device requirements are.

You can control virtual unit record devices with the CP SPOOL command. However, printed or punched output need not be printed or punched. For example, when using a technique for alternating between operating systems, you can spool the virtual printer to the card reader, as follows:

```
spool printer to *
```

Then after reIPLing CMS, the CMS RECEIVE command can be used to read printed output onto a CMS disk so that it can be examined. Also, use the SPOOL command to change the output spooling class of the virtual machine's spooled printer or punch files.

Defining the Operator's Console

During DOS system generation, an address is specified for the operator's console. The user's terminal must also be at this address. Usually, DOS expects the operator's console to be at real address 01F. The device has to be generated as a 3215, 3210, or 1052 console. However, when using DOS in a virtual machine, any terminal type can be used as the virtual operator's console.

Note: AF-DOS/VS and VSE/AF support 3270's as DOS operator consoles in full screen mode.

To find out the virtual machine's terminal address, enter the CP command:

```
query console
```

If the response indicates that the terminal is not at 01F but at another address, such as 009 (which is a standard VM/SP console address), enter this command:

```
define 009 as 01f
```

This command allows the terminal to now function as the operator's console for both DOS and CMS.

Preparing Jobs for a DOS Virtual Machine

There are several ways to prepare a job stream for a DOS virtual machine:

- Prepare a deck of punched cards that contains such information as DOS job control statements and input files. Place a CP ID statement at the beginning of this deck to indicate the userid of the DOS virtual machine. For example:

```
ID VSEAF
```

Then put the cards in the real system card reader. Based on the userid specified on the ID card, VM/SP directs the spool file to the virtual card reader of the DOS virtual machine, which in this case is being run on the virtual machine with a userid of VSEAF.

- Use CMS to create a disk file containing card images identical to the cards submitted in a real card reader for DOS. Use the CP SPOOL command to spool the virtual card punch to the card reader of the DOS virtual machine and use the CMS PUNCH command to punch the card images.

Before using the virtual punch to punch jobs to a virtual machine, take the precaution of clearing any files or card images that may remain in it from previous jobs. The following command ensure that the virtual punch does not have any other punch files in it:

```
spool punch nocont purge
```

In the CMS environment, issue:

```
spool punch to vseaf  
punch dosjob jcl (noheader
```

Use the NOHEADER option of the PUNCH command to suppress punching a CMS READ control card at the beginning of the card deck.

A job stream spooled to DOS by either of these methods remains in the card reader of the DOS virtual machine until the user causes DOS to begin reading the job stream from its card reader.

Spooling the card file can be done before or after initializing DOS or at any time while the DOS system is active.

You should also issue these commands to purge any existing reader files of the virtual machine that runs DOS:

```
spool reader nocont  
close reader  
purge reader
```

Of course, do not purge any needed files that may be in the reader. To obtain data about existing reader files before they are purged, enter the command:

```
query reader all (from USEAF)
```

Note: This command will not inform you if any of the files remain open.

Loading DOS

This topic describes three methods for loading DOS in a virtual machine. The first method uses the Automatic System Initialization (ASI) facility of VSE/AF and the second method uses the Saved System facility of VM/SP. The third method shows how to enter the commands and control statements to IPL DOS and how to ready DOS for input jobs.

IPL Using ASI

The VSE/AF Supervisor used in the examples was generated with VM Handshaking and runs in nonpaging mode. It is assumed that you are already logged on as userid VSEAF. The minimum storage size needed for this example is 6M. To show you the virtual hardware resources we have issued the **query virtual all** command. The response shows the storage size, device addresses, etc.

cp query virtual all

```
STORAGE = 06144K
CHANNELS = BMX
RDR 00C CL * NOCONT NOHOLD EOF READY
PUN 00D CL B NOCONT NOHOLD COPY 001 READY FORM STANDARD
    00D FOR VSEAF DIST 999/999/
PRT 00E CL A NOCONT NOHOLD COPY 001 READY FORM STANDARD
    00E FOR VSEAF DIST 999/999/ FLASHC 000
    00E FLASH CHAR PN MDFY 0 FCB 8
CONS 01F ON GRAF 057 TERM START
    01F CL T NOCONT NOHOLD COPY 001 READY FORM STANDARD
    01F FOR VSEAF DIST 999/999/ FLASHC 000
    01F FLASH CHAR TN MDFY 0 FCB 8
LINE 028 DISABLED
DASD 250 3330 VSERES R/W 404 CYL
DASD 251 3330 VSEPVT R/W 404 CYL
R;
```

By the **query terminal** command you can see that the virtual machine console is in printer/keyboard mode, in our example, "CONMODE 3215".

cp query term

```
LINEND # , LINEDEL ¢ , CHARDEL @ , ESCAPE " , TABCHAR ON
LINESIZE 080 , ATTN OFF , APL OFF , TEXT OFF , MODE VM ,
HILIGHT ON CONMODE 3215 , BREAKIN IMMED , BRKKEY PA1 ,
SCRNSAVE OFF
R;
```

The response to the **query set** command shows that S370 Extended Control mode is active for the virtual machine "ECMODE ON".

cp query set

```
MSG ON , WNG ON , EMSG ON , ACNT OFF , RUN OFF
LINEDIT ON , TIMER ON , ISAM OFF , ECMODE ON
ASSIST ON SVC TMR , PAGEX OFF , AUTOPOLL OFF
IMSG ON , SMSG OFF , AFFINITY NONE , NOTRAN OFF
VMSAVE OFF , 370E OFF
STBYPASS OFF , STMULTI 3/0 00/000
VMCONIO OFF , CPCONIO OFF
R;
```

We are ready to load VSE/AF into the virtual machine. This is done by IPLing 250. Because the Automatic System Initialization (ASI) facility is being used, no additional operator responses are required until the majority of the system initialization is complete.

The next example shows how we IPLed our virtual machine. Throughout the example you will find highlighted comments added to clarify points of interest.

cp ipl 250

```

OI04I IPLDEV=X'250,VOLSER=VSERES,CPUID=FF0142494341
OJ01I IPL=$IPL370 ,JCL=$$JCL370,SUPVR=$$A4SUPV,P
OI30I DATE=12/05/83,CLOCK=14/51/01,ZONE=EAST/00/00
    THE DATE VALUE FORMAT IS MM/DD/YY
ADD 00C,2540R      .CARD READER
ADD 00D,2540P      .CARD PUNCH
ADD 00E,1403U      .LINE PRINTER
ADD 01F,1050A      .SYSLOG
ADD 028,2701      .VSE/POWER BSC/RJE LINE
ADD 181:184,3420T9 .TAPE
ADD 250:257,3330   .SYSRES+WORK
DEF SYSREC=250,SYSCAT=251
OJ10I IPL RESTART POINT BYPASSED
SVA SDL=64,PSIZE=256K,GETVIS=512
OJ24I DASD SHARING SUPPORT RESET
OI26I $$BUCB4 LOADED          CUU=00E
OI20I IPL COMPLETE FOR VSE/AF 5746-XE8 REL 3.0 ECLEV=0000
    SUPVR USERID IS: VM/SP:SYSTEMTEST
BG 000 STDOPT CHARSET=60C,RLD=YES,SXREF=YES,SYM=YES,TERM=YES
BG 000 ALLOC F1=768K,F2=176K,F3=176K,F4=176K,F5=176K,F6=176K
BG 000 ALLOC F7=176K,F8=176K,F9=176K,FA=176K,FB=176K
BG 000 ALLOCR F1R=128K
BG 000 SIZE BG=512K,F1=512K,F2=128K,F3=128K,F4=128K,F5=128K
BG 000 SIZE F6=128K,F7=128K,F8=128K,F9=128K,FA=128K,FB=128K
BG 000 ASSGN SYSLST,00E
BG 000 ASSGN SYSPCH,00D
BG 000 1T20I  SYSLNK HAS BEEN ASSIGNED TO X'251'
BG 000 ASSGN SYS001,DISK,VOL=VSEPVT,SHR
BG 000 1T20I  SYS001 HAS BEEN ASSIGNED TO X'251'
BG 000 ASSGN SYS002,DISK,VOL=VSEPVT,SHR
BG 000 1T20I  SYS002 HAS BEEN ASSIGNED TO X'251'
BG 000 ASSGN SYS003,DISK,VOL=VSEPVT,SHR
BG 000 1T20I  SYS003 HAS BEEN ASSIGNED TO X'251'
BG 000 ASSGN SYS004,DISK,VOL=VSEPVT,SHR
BG 000 1T20I  SYS004 HAS BEEN ASSIGNED TO X'251'
BG 000 // OPTION STDLABEL
BG 000 // DLBL IJSYSRS,'VSEAF.SYSRES.FILE',99/365
BG 000 // EXTENT SYSRES,VSERES
--- (Here the rest of the System Standard Labels are set.)
BG 000 // OPTION PARSTD
BG 000 // DLBL IJSYSLN,'BG.SYSLNK',0
BG 000 // EXTENT SYSLNK,VSEPVT,,95,380          005-00 => 024-18
--- (Here the rest of the BG Partition Standard Labels are set.)
BG 000 // OPTION USRLABEL
BG 000 // PAUSE 'SET RF=CREATE AND/OR HC=CREATE' IF NECESSARY
BG-000
0    (A null response is given because the Recorder File is already formatted and the Hard Copy file is not used
    when running in PRT/KYBD mode.)
=>
BG 000 SET SDL
BG 000 ASSGN SYSLST,UA
BG 000 ASSGN SYSPCH,UA
BG 000 START F1
BG 000 STOP

```

Figure 3-1 (Part 1 of 3). IPLing DOS Using Automatic System Initialization (ASI)

```

F1 001 ASSGN SYSLST,00E
F1 001 ASSGN SYSPCH,00D
F1 001 ASSGN SYSLNK,DISK,VOL=VSEPVT,SHR
F1 001 1T20I SYSLNK HAS BEEN ASSIGNED TO X'251'
F1 001 ASSGN SYS000,UA No job accounting
F1 001 ASSGN SYS001,DISK,VOL=VSEPVT,SHR IJQFILE
F1 001 1T20I SYS001 HAS BEEN ASSIGNED TO X'251'
F1 001 ASSGN SYS002,DISK,VOL=VSEPVT,SHR IJDFILE
F1 001 1T20I SYS002 HAS BEEN ASSIGNED TO X'251'
F1 001 // JOB VSEPOWER with RJE

DATE 12/05/83,CLOCK 14/51/31
F1 001 1I93I RECORDER FILE IS 1% FULL
F1 001 // OPTION NODUMP
F1 001 // PAUSE Enter /+ to bypass VSE/POWER autostart---
F1-001
1
=>
F1 001 // EXEC POWERRJE
F1 001 1QB7I QUEUE FILE RECOVERY IN PROGRESS
F1 001 1QB8I QUEUE FILE RECOVERY COMPLETED
F1 001 1Q20I AUTOSTART IN PROGRESS
F1 001 1R75I BG AUTOSTARTED
F1 001 1R75I F2 AUTOSTARTED
F1 001 1R75I F3 AUTOSTARTED
F1 001 1R75I F4 AUTOSTARTED
F1 014
OP08A INTERV REQ SYS000=00C (This message indicates there are no files in VSEAF's virtual reader.)
F1 001 1R75I F5 AUTOSTARTED
F1 001 1R75I F6 AUTOSTARTED
F1 001 1R75I F7 AUTOSTARTED
F1 001 1R75I F8 AUTOSTARTED
F1 001 1R75I F9 AUTOSTARTED
F1 001 1R75I FA AUTOSTARTED
F1 001 1R75I FB AUTOSTARTED
F1 001 1Q34I LST WAITING FOR WORK ON 00E
F1 001 1Q34I PUN WAITING FOR WORK ON 00D
F1 001 1Q12I VSE/POWER INITIATION COMPLETED
F2 002 ASSGN SYSLST,00E
F2 002 ASSGN SYSPCH,00D
F2 002 ASSGN SYSLNK,DISK,VOL=VSEPVT,SHR
F2 002 1T20I SYSLNK HAS BEEN ASSIGNED TO X'251'
F2 002 ASSGN SYS001,DISK,VOL=VSEPVT,SHR
F2 002 1T20I SYS001 HAS BEEN ASSIGNED TO X'251'
F2 002 ASSGN SYS002,DISK,VOL=VSEPVT,SHR
F2 002 1T20I SYS002 HAS BEEN ASSIGNED TO X'251'
F2 002 ASSGN SYS003,DISK,VOL=VSEPVT,SHR
F2 002 1T20I SYS003 HAS BEEN ASSIGNED TO X'251'
F2 002 ASSGN SYS004,DISK,VOL=VSEPVT,SHR
F2 002 1T20I SYS004 HAS BEEN ASSIGNED TO X'251'
F2 002 // OPTION PARSTD
F2 002 // DLBL IJSYSLN,'F2.SYSLNK',0
F2 002 // EXTENT SYSLNK,VSEPVT,,,1995,19 105-00 => 105-18
F2 002 // DLBL IJSYS01,'F2.SYS001',0
F2 002 // EXTENT SYS001,VSEPVT,,,2014,19 106-00 => 106-18
F2 002 // DLBL IJSYS02,'F2.SYS002',0
F2 002 // EXTENT SYS002,VSEPVT,,,2033,19 107-00 => 107-18
F2 002 // DLBL IJSYS03,'F2.SYS003',0
F2 002 // EXTENT SYS003,VSEPVT,,,2052,19 108-00 => 108-18
F2 002 // DLBL IJSYS04,'F2.SYS004',0
F2 002 // EXTENT SYS004,VSEPVT
F2 002 // OPTION USRLABEL

```

Figure 3-1 (Part 2 of 3). IPLing DOS Using Automatic System Initialization (ASI)

```
F2 002 EOP $2JCL370
F2 002 // ASSGN SYSIN,00C,PERM
F1 001 1Q34I F2 WAITING FOR WORK
```

--- (At this point the rest of the partitions are brought up under VSE/POWER.)

--- (The responses to the following DOS and POWER operator commands show the status of the DOS system after system initialization is completed.)

map

```
=>
AR 015          AREA          SIZE  GETVIS  REAL  UPPER-LIMIT  NAME
AR 015          SP            200K          200K    31FFF    $$$
AR 015          BG  ACV       512K    1044K    OK     1B6FFF    NO NAME
AR 015          FB  ABV       128K     48K     OK     1E2FFF    NO NAME
AR 015          FA  AAV       128K     48K     OK     20EFFF    NO NAME
AR 015          F9  A9V       128K     48K     OK     23AFFF    NO NAME
AR 015          F8  A8V       128K     48K     OK     266FFF    NO NAME
AR 015          F7  A7V       128K     48K     OK     292FFF    NO NAME
AR 015          F6  A6V       128K     48K     OK     2BEFFF    NO NAME
AR 015          F5  A5V       128K     48K     OK     2EAFFF    NO NAME
AR 015          F4  A4V       128K     48K     OK     316FFF    NO NAME
AR 015          F3  A3V       128K     48K     OK     342FFF    NO NAME
AR 015          F2  A2V       128K     48K     OK     36EFFF    NO NAME
AR 015          F1  A1V       512K    256K    128K    42EFFF    NO NAME
AR 015          SVA A         980K    880K          5FFFFFF    VSEPOWER
AR 015          PP                                OK
```

d q (display POWER queue information)

```
=>
F1 001 1R49I  FREE RECORDS QUEUE FILE 227
F1 001 1R49I  NO ACCOUNTING SUPPORT
```

d t (display time)

```
=>
F1 001 1R46I  TIME IS 14:52:45, DATE IS 12/05/83
F1 001 1R46I  014 PAGES FIXED, 017 CURRENT TASKS
```

d a (display active)

```
=>
AR 015 0P69I  INTERV REQ  F1 00C
F1 001 1R48I  LST,00E,A,1 INACTIVE
F1 001 1R48I  PUN,00D,A, INACTIVE
F1 001 1R48I  BG,00C,0, INACTIVE
F1 001 1R48I  F2,00C,2, INACTIVE
F1 001 1R48I  F3,00C,3, INACTIVE
F1 001 1R48I  F4,00C,4, INACTIVE
F1 001 1R48I  F5,00C,5, INACTIVE
F1 001 1R48I  F6,00C,6, INACTIVE
F1 001 1R48I  F7,00C,7, INACTIVE
F1 001 1R48I  F8,00C,8, INACTIVE
F1 001 1R48I  F9,00C,9, INACTIVE
F1 001 1R48I  FA,00C,A, INACTIVE
F1 001 1R48I  FB,00C,B, INACTIVE
F1 001 1R48I  RDR,00C,A,
```

#cp savesys vseaf --- (The SAVESYS saves an IPLable DOS system at the then-current status of the virtual machine. In other words, what you save is what you get back when you IPL the Saved System. See the *VM/SP Planning Guide and Reference* for additional information on defining and using a saved system.)

SYSTEM SAVED

Figure 3-1 (Part 3 of 3). IPLing DOS Using Automatic System Initialization (ASI)

IPLing Using a Saved System

By IPLing a saved system, you are able to bypass all of the DOS system initialization steps as shown in the first example. However, you need to be aware that by IPLing VSEAF you get an exact copy of your DOS system that was previously saved. If you want your saved system copy of DOS system at the end of its last session, you may wish to use the VMSAVE facility of VM/SP.

```
cp ipl vseaf
d a

=>
AR 015 OP69I INTERV REQ F1 00C
F1 001 1R48I PUN,00D,A,INACTIVE
F1 001 1R48I LST,00E,A,1 INACTIVE
F1 001 1R48I BG,00C,0,INACTIVE
F1 001 1R48I F2,00C,2,INACTIVE
F1 001 1R48I F3,00C,3,INACTIVE
F1 001 1R48I F4,00C,4,INACTIVE
F1 001 1R48I F5,00C,5,INACTIVE
F1 001 1R48I F6,00C,6,INACTIVE
F1 001 1R48I F7,00C,7,INACTIVE
F1 001 1R48I F8,00C,8,INACTIVE
F1 001 1R48I F9,00C,9,INACTIVE
F1 001 1R48I FA,00C,A,INACTIVE
F1 001 1R48I FB,00C,B,INACTIVE
F1 001 1R48I RDR,00C,A

map
=>
AR 015 OP69I INTERV REQ F1 00C
AR 015 AREA SIZE GETVIS REAL UPPER-LIMIT NAME
AR 015 SP 200K 200K 31FFF $$A$$SUPV
AR 015 BG ACV 512K 1044K OK 1B6FFF NO NAME
AR 015 FB ABV 128K 48K OK 1E2FFF NO NAME
AR 015 FA AAV 128K 48K OK 20EFFF NO N
AR 015 F9 A9V 128K 48K OK 23AFFF NO NAME
AR 015 F8 A8V 128K 48K OK 266FFF NO NAME
AR 015 F7 A7V 128K 48K OK 292FFF NO NAME
AR 015 F6 A6V 128K 48K OK 2BEFFF NO NAME
AR 015 F5 A5V 128K 48K OK 2EAFFF NO NAR
AR 015 F4 A4V 128K 48K OK 316FFF NO NAME
AR 015 F3 A3V 128K 48K OK 342FFF NO NAME
AR 015 F2 A2V 128K 48K OK 36EFFF NO NAME
AR 015 F1 A1V 512K 256K 128K 42EFFF VSEPOWER
AR 015 SVA A 980K 880K 5FFFFF
AR 015 PP OK
```

Figure 3-2. IPLing DOS Using the Saved System Facility of VM/SP

The responses to the DOS and POWER operator commands shows that the IPLed saved system copy of DOS is exactly the same as when the system was saved.

IPL from the Console

In order for paging to occur while using DOS, the Extended Control Mode (EC) must be set on. Before this can be done you must be sure of the following:

- All the virtual unit record devices needed are attached
- The console has been properly defined
- All required DASD units are attached

Now, you can set the Extended Control Mode on by typing this command:

```
set ecmode on
```

You can avoid having to enter the ECMODE command if you specify it in your virtual machine's directory.

To load DOS into the virtual machine, enter the IPL command:

```
ipl 250
```

After a few seconds, the system enters a wait state. On a real system console, the wait light goes on. On the terminal, a user may want to let a few seconds pass to be sure that the wait state has been entered. To verify that the system is in a wait state, enter CP mode and use the DISPLAY command to display the PSW:

```
display psw
```

If bit 14 is a 1, the system is in a wait state. If not, use the BEGIN command to resume program execution.

After determining that the system is in a wait state, cause an attention interruption by pressing the attention key (or equivalent). The following message asks you to enter the name of the DOS supervisor:

```
OIO3A SPECIFY SUPERVISOR NAME
```

Entering a null line causes DOS to use the default supervisor, \$\$A\$SUP1.

Shortly after entering the supervisor name, the system enters the wait state again. Allow a few seconds to pass to be sure that the wait state has been reached. Then, cause another attention interruption with the attention key (or its equivalent).

DOS displays a series of messages that indicates the status of the IPL procedure, followed by a prompting message:

```
OIO10A GIVE IPL CONTROL COMMANDS
```

In response to this message, enter the following commands in the given order:

1. The ADD or DEL command (optional) to alter the default DOS configuration that is established at system generation.
2. The SET command (required) to initialize the date and time clock.
3. The CAT command (optional) to define the VSAM master catalog.
4. The DPD command (required) to define the page data set.

After entering the DPD command, this message appears:

```
OIO20I DOS IPL COMPLETE
```

indicating that DOS is loaded into the virtual machine.

If a warm start copy of the SVA (shared virtual area) is available, this message will also appear:

```
1T00A WARM START COPY OF SVA FOUND
```

In response, enter **KEEP** (or a null line) or **REJ**, depending upon whether this copy of the SVA is to be used.

If no warm start copy of the SVA is available and the SVA must be used, create one by using a standard procedure, depending upon what is available in DOS.

When the IPL procedure is complete, this message appears:

```
BG  
1I00A READY FOR COMMUNICATIONS
```

It indicates that the background partition is running and is ready to accept control commands or job control statements.

The complete VM/SP logon and DOS IPL procedure as it would appear on a 3270 terminal is shown in Figure 3-3 on page 3-16. The exclamation marks (!) indicate pressing the attention key (or its equivalent).

Note: If issuing the **SET HC=CREATE** command, the console must be in display mode. See “Specifying Virtual Machine Consoles” in Section 1 or “Defining a console for VM/SP in a Virtual Machine” in Section 2 for further information on defining a console.

logon tester

ENTER PASSWORD:

FILES: 001 RDR, NO PRT, NO PUN (see Note 1)
LOGON AT 08:19:52 EST MONDAY 01/23/83

set ecmode on

link dosys 250 250 w (see Note 2)

DASD 250 LINKED R/W

ipl 250 (see Note 3)

!
OI03A SPECIFY SUPERVISOR NAME

\$\$\$sup2

!
OI04I IPLDEV=X'250',VOLSER=DOSRE3,CPUID=FF0101530155

OI30I DATE=01/23/83,CLOCK=16/35/09,ZONE=EAST/04/00

OI10A GIVE IPL CONTROL COMMANDS

set (see Note 4)

dpd

OI52I PAGE DATA SET EXTENT LOW HIGH
 250 0 327 11

OI20I DOS IPL COMPLETE

BG

1I00A READY FOR COMMUNICATIONS.

BG

(see Note 5)

log

BG

assgn sysrec,x'250'

BG

set rf=yes, SET HC=yes

BG

(see Note 6)

BG

// JOB TEST1

DATE 01/23/83,CLOCK 16/35/47

BG

1I89A IPL REASON CODE =

(see Note 7)

BG

1I91A SUB-SYSTEM ID =

BG

1I93I RECORDER FILE IS 2% FULL

BG

(see Note 8)

// OPTION NODECK

BG

// EXEC ASSEMBLY

BG

EOJ TEST1

DATE 01/23/83,CLOCK 16/37/49,DURATION 00/02/01

BG

1C00A ATTN. 00C

(see Note 9)

BG

(see Note 10)

BG

OP08A INTERV REQ SYSRDR=00C

Figure 3-3. IPL DOS and Execute a Job in the Card Reader

Notes:

1. The reader file contains a job stream for the DOS virtual machine.
2. The DOS system residence is assumed to be defined at virtual address 250 in userid DOSYS's directory entry and having a write password of ALL.
3. After the IPL command, the attention interruption causes message 0I03A to be issued. After entering the supervisor name, a later interruption continues the IPL procedure.
4. The SET and DPD commands are required to IPL DOS.
5. The background partition is active and waiting for commands. The LOG command is optional; the ASSGN and SET commands shown here may be required for system recording; the optional SET HC= command shown specifies hardcopy output, but the console must be in display mode.
6. A null line signals an interruption to the card reader, so that DOS begins reading the job stream.
7. These are RDE (reliability data extractor) messages; a null line entered in response indicates that you are taking the default values.
8. DOS continues reading the job.
9. The card reader is empty; the spool file has been read.
10. A null line causes another interrupt to the card reader; if another file is in the card reader, it is read. Otherwise, message 0P08A is issued.

IPL from the Card Reader

Before you issue the IPL command to load the system, place the control commands for performing the IPL procedure in the card reader of the DOS virtual machine. Then, signal DOS to read the control commands from the card reader. This method can be more efficient than entering all of the control commands manually, especially if there are many ADD and DEL commands that must be issued when initializing DOS.

The cards required to perform the IPL procedure must be in two separate card files: (1) a single card specifying the supervisor name, and (2) a card deck containing the IPL commands. A user may follow these card files with additional files that contain jobs for execution in the DOS virtual machine.

To load DOS into the virtual machine, enter the IPL command:

```
ipl 250
```

When the system enters a wait state, IPL DOS from the card reader causing a reader interruption. This interruption allows the system to read the name of the supervisor from the card reader. To cause a reader interruption, enter the CP environment by pressing the attention key twice (or the 3270's PA1 key once) and enter these commands:

```
#cp ready 00c  
#cp begin
```

The READY command causes a reader interruption; the BEGIN command returns control to DOS which then reads the first spool file from the card reader.

Shortly after this card file is read, the system enters the wait state a second time. You must again enter the CP environment and enter these commands:

```
ready 00c
begin
```

In response to these commands, DOS begins reading the IPL commands.

Note: When initializing DOS through the card reader, you cannot supply the responses necessary to save the warm start copy of the SVA from card input. These responses must be supplied from the console. To create the SVA and SDL during IPL, place the cards in the card reader, but in a separate spool file following the IPL commands. Again, you must enter the READY 00C command to cause the reader interruption to force the system to begin reading from the card reader.

DOS Operation

Depending upon how DOS was generated, there may be additional operator commands and control statements that must be entered at the console running jobs on the DOS virtual machine.

If there is an active system recorder file, it is opened when the first // JOB card is encountered in the input stream. Before this JOB card is read, enter the ASSGN and SET commands that define the SYSREC device and the status of the system recorder file. Otherwise, an error is encountered opening the SYSREC file. For example, if the system recorder file is already active on the system residence volume, enter:

```
assgn sysrec,x'250' (for DOS Release 34 or earlier only)
set rf=yes
```

When starting a DOS virtual machine to run in batch mode to process jobs from other users, you may also want to enter the operator commands necessary to:

- Allocate storage among different partitions
- Start foreground partitions in operation
- Initialize POWER

These considerations are discussed later in this section under the topic "Running Batch DOS under VM/SP". When alternating between CMS and DOS, you may want to keep the IPL procedure as simple as possible.

Starting a Job Stream (W/O POWER)

After preparing job streams and placing them in the DOS card reader, the message READY FOR COMMUNICATIONS appears. Enter a null line (with the return or enter key) to cause an interruption. This interruption causes DOS to begin reading from the card reader.

If the LOG command was entered before beginning the job stream, DOS displays on your terminal all the job control statements that are executed.

As the DOS operator, the virtual machine user can enter control statements directly from the terminal. For example, to run cataloged programs or procedures, enter the ASSGN, DLBL, and EXEC statements necessary to run the program directly from the console.

When a Job Is Finished

When the DOS virtual machine running a single partition finishes reading a spool file (that may contain one or more jobs), it posts this attention message to indicate that the reader is empty:

```
1C00A  ATTN. 00C
```

If additional jobs are in the card reader and were spooled as separate CP spool files, enter a null line to cause DOS to begin reading from the card reader. If no more files are in the reader and the user enters a null line, this message appears to indicate that operator intervention is required:

```
0P08A  INTERV REQ SYSRDR=00C
```

If desired, you can put cards into the real system card reader to direct another job stream to the DOS virtual machine. When the DOS virtual machine receives cards in its reader, it begins reading them automatically.

If the card reader of the DOS virtual machine is spooled with the CONT operand, then any jobs received in the card reader (while a job is running) are read upon completion of the current job stream. If a job stream completes and the end of the spool file is reached before another job is received, an interruption must be issued to cause the next job to be read.

To end the DOS terminal session, use the attention key (or equivalent) to enter the CP command environment. Under CP, either enter the LOGOFF command to log off the VM/SP system or use the IPL command to load another operating system into the virtual machine.

Communicating with CP

While operating the DOS virtual machine, use CP commands to:

- Communicate with the VM/SP system operator or other virtual machine users
- Query the status of virtual machine devices or spool files
- Attach or detach devices from the virtual machine configuration

If your DOS console is in printer/keyboard (3215) mode, you can communicate with CP by either:

- Pressing the ATTN key twice (or its equivalent) to force a CP read

----- or -----

- Preface the CP command with '#CP' where '#' is assumed to be the terminal linend character.

If your DOS console is in display (3270) mode then you can only communicate with CP by pressing the TERMINAL BRKKEY KEY (usually the PA1 key).

Notes:

1. Unless the SET RUN ON command is in effect, you will have to issue the CP BEGIN command to return control to your guest operating system.
2. If your DOS console is in display mode, it is recommended that you issue the CP TERMINAL SCRNSAVE ON command for your machine.

Example: If a DOS PAUSE control statement requests the VM/SP operator to perform some action, enter the CP environment to send a message to the VM/SP system operator. The following lines represent a typical sequence on a typewriter terminal (assuming this is BG partition and the DOS Asynchronous Operator Facility was not specified for this DOS system):

```
// PAUSE ASK OPERATOR TO ATTACH TAPE AS 284
#cp msg op please attach scratch tape as 284
begin
stop
TAPE 284 ATTACHED
start <NULL>
```

Note: At times, CP commands cannot be entered with the #CP function. For example, during the IPL procedure when DOS is processing IPL commands, the CCWs used for these reads expect only three bytes of data. Any additional information on CP command lines is truncated.

Interrupting the Virtual Machine

While DOS is running in the virtual machine, you can interrupt its execution by using the attention key (or its equivalent) on the terminal. When this key is pressed, the DOS attention handler responds with these messages:

```
AR
1I60A  READY FOR COMMUNICATIONS
AR
```

When these messages appear, you can enter attention commands. To resume program execution, enter a null line. Wait until the attention has been processed before signaling another one, except when cancelling a dump.

When using a 2741 terminal and its attention key mainly to signal CP interruptions, enter the command:

```
#cp terminal mode cp
```

The first time the attention key is pressed, VM/SP posts a CP interruption. The next pressing of the attention key signals an interruption to DOS. When you are in the CP environment and want to signal an interruption to DOS, enter either the ATTN or REQUEST commands.

Running Batch DOS Under VM/SP

When using DOS in a virtual machine as a production tool, it is likely that the virtual machine running DOS is going to be logged on continuously. This machine may be available for many users to submit jobs, or it may be used only by personnel responsible for running the production jobs.

In either event, it is likely that DOS has been generated specifically for use under VM/SP. In this case, you should know whether:

- It is necessary to start more than one partition. If you do, then you must determine how much virtual storage to allocate to each partition. These considerations are much the same as they would be for a native DOS user who must decide how much work is going to be done and the most efficient way of doing it.
- The POWER program provides many functions that are not available when relying solely on CP spooling, such as spooling jobs via class and purging selected jobs in a job stream.

When the POWER program is active in the DOS virtual machine, it controls which jobs are to be processed in the various partitions, and it reads jobs from the card reader. Because the program is constantly working, there is no need for an attention interruption when a new job is placed in a card reader.

To start the POWER program in a virtual machine, use the AUTOSTART procedure. Enter the POWER control statement for the automatic start, using a card reader or data in a DOS procedure.

- Virtual devices are dedicated for use by the DOS virtual machine or devices must be shared among other virtual machines.

Example: If a card reader has been dedicated to a DOS virtual machine, then users may submit jobs through that card reader. They do not have to place a CP ID card at the beginning of the deck to direct it to the appropriate virtual machine. If the DOS virtual machine is sharing the system card reader, then any user submitting a card deck must place a CP ID card at the beginning of the deck, specifying the userid of the DOS virtual machine.

Alternating Between CMS and DOS Under VM/SP

When working in a development and testing environment (rather than in a batch environment) and the work is not suitable for CMS/DOS, there are some advantages to alternating between CMS and DOS:

- Reduced unit record output. Under CMS, users can examine program output and compiler listings online, check the results and resubmit the job without printing a single page on the system printer or punching card decks.
- Faster turnaround time compared with batch alone. Under CMS users can see the results of program compilation and execution immediately, rather than waiting for output from a batch system.

This topic outlines the technique for alternating between operating systems. Before using this technique, you should be familiar with CMS, particularly with the System Product Editor and some file system commands. CMS usage information is in the *VM/SP CMS User's Guide*. For details on CMS commands, refer to *VM/SP CMS Command and Macro Reference*.

Loading CMS into a Virtual Machine

To load CMS into a virtual machine, use the IPL command. Usually, IPL CMS by specifying the saved system name CMS:

```
ipl cms
```

Or, load CMS by specifying the virtual address of the CMS system, usually 190:

```
ipl 190
```

After receiving a message like this:

```
CMS VM/SP 3.0
```

A user can use the interactive facilities of CMS to prepare jobs for execution in the DOS virtual machine.

Using the System Product Editor

Use the XEDIT command in CMS to pass control to the System Product Editor for preparing disk files of 80-character card image records. The created files may contain DOS job control statements, source files, or even IPL decks.

For example, to IPL DOS using input statements from the card reader, prepare CMS files that contain:

- The DOS supervisor name required for an IPL, such as the record:

```
$$A$SUP1
```

- The IPL control statements that a user wants to supply, such as:

```
set
dps
assgn sysrec,x'250'
set rf=yes
log
```

Assume for the purposes of this example that: (1) these files have CMS file identifications of SUPER JCL and START JCL, and (2) these files contain all the statements needed to control DOS IPL.

The System Product Editor can also be used to prepare the job stream selected for execution. For example, to assemble an assembler language source program and make the source file available as a CMS disk file, xedit the file and insert the DOS job control statements into the CMS file. The file DOSTEST JCL might contain:

```
// JOB TEST1
// OPTION NODECK
// EXEC ASSEMBLY
   (assembler language source statements)
   .
   .
/*
/ε
```

Although this job stream contains only the job control statements necessary to assemble the job, it can also include the statements necessary to link-edit the output module, catalog the program, and so on.

Issuing SPOOL Commands To Control Unit Record Devices

Before sending a job to the DOS virtual machine, check the unit record devices to make sure that no files are left over from a previous job. Close and purge the card reader.

```
spool reader nocont
close reader
purge reader all
```

Close and purge the card punch, and spool it to the card reader:

```
spool punch nocont
close punch purge
spool punch to *
```

When using the card reader to IPL DOS, spool the punch NOCONT. At least the first two files must be separate spool files.

Spool the virtual printer to the card reader:

```
spool printer to *
```

For DOS SYSLST output to print on the system printer, omit this SPOOL command. You can also spool the printer file to a different class, such as L. This action ensures that the printer is closed before you are ready to IPL CMS and that DOS does not read the printer file from the reader:

```
spool printer to * class L
```

Punching CMS Files

Use the CMS PUNCH command to punch the card files. The files are spooled to the virtual card reader. For example, to punch the three files used above enter:

```
punch super jcl (noheader
punch start jcl (noheader
punch dostest jcl (noheader
```

The NOHEADER option must be used to suppress punching a CMS READ control card, that is punched by default when using the PUNCH command.

Initializing DOS

When the DOS system residence volume is attached to the virtual machine, use the IPL command to load DOS:

```
ipl 250
```

After waiting a few moments, use the DISPLAY command to see if the system is in a wait state. After determining that the system is in a wait state, enter an attention interruption:

```
!!
CP
display psw
PSW = 030E000 00000000
ready 00c
begin
```


After the BEGIN command returns control to the virtual machine, the file SUPER JCL is read from the card reader, and the IPL procedure continues. Again, you must wait for the system to enter a wait state and ready the reader:

```
!!
CP
display psw
PSW = 030E0000 00012800
ready 00c
begin
```

When the IPL procedure is complete, messages such as these appear:

```
0I52I PAGE DATA SET EXTENT      LOW      HIGH
                                250    11    327    0
0I20I IPL COMPLETE FOR DOS REL xx.x ECLEVEL=nn
BG
1C00A  ATTN. 00C
BG
```

The first message DOS MSG0I52I will not appear if your DOS system is running in nonpaging mode.

As a practical matter, you may want to create a DOS saved system at this point so that the future loading of DOS would avoid these preliminary steps. Refer to the *VM/SP System Programmer's Guide* for information on creating saved systems.

Signaling DOS To Read the Job Stream

When DOS has been loaded and the message BG indicates that it is waiting for communication, enter a null line. This action causes DOS to begin reading the next spool reader file. This file is the one that contains the job control statements, and in this example, the assembler language source program.

If the LOG command is issued during the IPL procedure, all of the job control statements that are read are displayed on the terminal as they are executed.

Note: If the RDE option is in effect for DOS Release 34 (or earlier), respond to these messages before the job is executed:

```
1I89A  IPL REASON CODE =
1I91A  SUB-SYSTEM ID =
```

When the Card Reader Is Empty

When the job sent from the punch to the card reader has been read, this message appears:

```
EOJ TEST1
```

The above message is followed by the time stamp:

```
DATE 02/19/83,CLOCK 20/19/28 DURATION 00/07/48
```

Additional messages from the background partition may also appear:

```
BG
1C00A ATTN. 00C
BG
```

When punching more than one job and the spool files for the remaining jobs are still in the card reader, enter another null line to signal DOS to read from the reader.

Reloading CMS into a Virtual Machine

When a job (or jobs) is finished in the DOS virtual machine, the user may return to the CMS environment by entering CP mode and issuing:

```
ipl cms
```

This IPL command closes the virtual printer. If the printer is spooled to the card reader or if the printer is in hold status, this message from VM/SP appears:

```
PRT FILE 4786 TO BILBO COPY 01 NOHOLD
```

----- or -----

```
PRT FILE 4786 FOR BILBO COPY 01 HOLD
```

This printer file contains the SYSLST output from the job that executed in DOS.

If the file is spooled to the printer, it is queued for printing on the system printer. If the printer is spooled to the card reader, it is a reader file that is available to the user in the card reader.

Examining DOS Virtual Machine Output

To examine a job's output executed under the DOS virtual machine, issue the RL (reader list) command to list the file(s) in your reader. Place the cursor under the appropriate line, and press PF 11 (peek) to look at the file.

When you examine assembly or compilation output, use the CLOCATE subcommand to locate a keyword in the listing file. For example, to locate the beginning of the diagnostic messages produced by the assembler, enter the following subcommand and press PF 5.

```
clocate/diagnostics
```

If no errors are in the assembly, modify the job control statements in the file and resubmit the job. Otherwise, correct the source statements and then start over. Starting over means that a user must repeat the procedure for clearing files from the card punch and card reader, punching the IPL decks and the job stream, and reinitializing DOS.

Using EXEC Procedures

When alternating extensively between CMS and DOS in a virtual machine, place in an EXEC procedure the commands necessary to spool unit record devices, punch the IPL deck, and punch a job stream. Then, when sending a job to the card reader and initializing DOS, you only have to enter the EXEC name and, in some cases a few additional control commands.

The following is an example of an EXEC procedure as mentioned above:

```
&CONTROL OFF
CP SPOOL PUNCH NOCONT
CP CLOSE PUNCH PURGE
CP CLOSE C
CP PURGE READER ALL
PUNCH SUPER JCL (NOH
PUNCH START JCL (NOH
PUNCH DOSTEST JCL (NOH
CP IPL 250
```

If the preceding EXEC procedure was named DOSJOB, to execute you would just enter:

```
dosjob
```

Once the IPL command in the EXEC is performed, the virtual machine is no longer under the control of CMS. To begin DOS operation, enter attention interruptions and CP commands, as necessary.

You can make an EXEC procedure more generalized, thereby making it capable of sending any job to the card reader. Use the EXEC symbol &1 in place of the DOSTEST filename, see below:

```
&CONTROL OFF
CP SPOOL PUNCH NOCONT
CP CLOSE PUNCH PURGE
CP CLOSE C
CP PURGE READER ALL
PUNCH SUPER JCL (NOH
PUNCH START JCL (NOH
PUNCH &1 JCL (NOH
CP IPL 250
```

Using the above EXEC you can send any job to the card reader as long as it has a CMS filetype of JCL. If you were to enter:

```
dosjob prog3
```

The EXEC variable symbol &1 is replaced with the argument PROG3 and the file PROG3 JCL is punched to the card reader for execution under DOS.

As an additional aid for entering the commands and responses necessary to IPL DOS from the card reader, use the CONT option of the SPOOL command. This option allows a user to spool the IPL commands and job stream as a single spool file. For example:

```
&CONTROL OFF
CP SPOOL PUNCH NOCONT
CP CLOSE PUNCH PURGE
CP CLOSE C
CP PURGE READER ALL
PUNCH SUPER JCL (NOH
CP SPOOL D CONT
PUNCH START JCL (NOH
PUNCH DOSTEST JCL (NOH
CP SPOOL D NOCONT
CP IPL 250
```

When spooling the punch with the CONT operand, VM/SP spools the START JCL and DOSTEST JCL files as a single spool file. When the IPL commands are read, DOS continues reading from the card reader. You do not have to signal (with a null line) when DOS is to begin reading the job stream.

Continuous spooling has an additional advantage in the IPL procedure. When issuing the IPL command, the virtual punch is closed. However, in the previous example where the punch was spooled NOCONT, but not closed, the IPL closed the punch. As the file is spooled to the reader, this action causes a reader interruption; this is the first interruption that you have to enter (the one that causes DOS to read the supervisor name).

Thus, when executing this EXEC format, you are required (after the IPL command is executed) to enter only one interruption and one READY command to execute the entire job. The console sheet might look like this:

```
dosjob
!!
display psw
PSW = 030E0000 000128000
ready 00c
begin
```

After issuing the BEGIN command, the remainder of the job is executed without user intervention until the card reader is empty.

The CMS EXEC facility is described in detail in the *VM/SP CMS User's Guide*.

Using More Than One Virtual Machine

In a DOS virtual machine, if you are running a job that may take a long time to execute, you may want to free your terminal for other work. In these situations, use the DISCONN command to disconnect the DOS virtual machine and logon to some other userid.

This topic shows a sample procedure (using the userids CMSPREP1 and DOSTEST1), followed by a list of some additional items that must be considered when disconnecting a terminal.

Note: While it is not necessary to use CMS to prepare or transmit the jobs to the DOS virtual machine, you may find it convenient.

Accessing CMS

To access CMS, you must first have a userid. In the example that follows the highlighted lines are the commands you type in. You can logon and load CMS as follows:

```
logon cmsprep1
ENTER PASSWORD:

LOGON AT 10:30:41 EST TUESDAY 02/19/83
ipl cms
CMS VERSION 3.0
```

Assuming that you have a read/write CMS A-disk, you can create CMS files that contain the IPL control commands and responses, as well as the job streams to execute in DOS. Then, by using the SPOOL and PUNCH commands, place copies of these files in the card reader of the machine that is going to run DOS.

Example: You have two files named SUPER JCL and START JCL, which contains the supervisor name and IPL commands for the IPL procedure. In addition, the file DOSTEST JCL contains the job stream that you want to execute. To spool these files to the userid DOSTEST1, enter:

```
cp spool punch to dostest1
punch super jcl (noheader
punch start jcl (noheader
punch dostest jcl (noheader
```

Disconnecting CMS

When you are ready to logon to the userid that is going to run DOS, disconnect the CMS virtual machine as follows:

```
disconn hold
```

You should use the HOLD operand of the DISCONN command when you use a dial-up terminal and do not want to lose the connection.

Since the CMS machine is not currently active, there is no need to disconnect it. While you could just as easily log off, disconnecting and logging on again saves reloading CMS or respooling the punch, printer, and so on.

Logging Onto DOS

When logging onto the virtual machine that is going to run DOS, in addition to the normal log messages, you see a message indicating that there are files in the card reader:

```
logon dostest1
.
.
.
FILES: 003 RDR, NO PRT, NO PUN
LOGON AT 10:50:34 EST TUESDAY 02/19/83
```

If this virtual machine does not have the ECMODE option set on (determined by issuing the QUERY SET command), issue the command SET ECMODE ON. Then, begin the IPL procedure as outlined under the topic "IPL from the Card Reader" (described previously in this section).

Note: If the RDE option is in effect for DOS Release 34 (or earlier), wait until after responding to the RDE messages before disconnecting the DOS virtual machine. Then, enter CP mode using the attention key, and enter the SET RUN ON and DISCONN commands.

```
!!  
CP  
set run on  
disconn
```

The DOS virtual machine continues to run.

Returning to CMS

You can now use the VM/SP logon procedures to reconnect to the CMS virtual machine. During logon, this message appears instead of the normal LOGON message:

```
RECONNECT AT 11:05:02 EST TUESDAY 02/19/83
```

To return to the CMS environment and continue working, enter the BEGIN command.

To see if the virtual machine that is set up to run DOS is running, issue this command:

```
query dostest1
```

After issuing the above command, you can continue to use CMS (or some other operating system) in this virtual machine.

If desired, you can again disconnect the CMS virtual machine and reconnect onto the DOS virtual machine. For example, you know that the job stream being executed may pause and require an operator response. Or, you may use CMS to spool another job to DOS. Then, you may need to reconnect to the DOS virtual machine to alert DOS to begin reading from the card reader.

After reconnecting, issue the BEGIN command for the DOS virtual machine to resume execution.

Note: By using the single console image facility described in the *VM/SP System Programmer's Guide*, both the CMS virtual machine and the DOS virtual machine can run concurrently from the same terminal. There is no need for repeated disconnecting and reconnecting of these virtual machines.

Disconnection Considerations

When using more than one userid to alternate between two operating systems, consider:

- How DOS may read additional jobs from the card reader.
- What happens when there is a read from the console of the DOS virtual machine.
- What happens to the console log of a disconnected virtual machine.
- If the single console image facility could be useful in controlling multiple virtual machines concurrently from one terminal.

Sending Jobs to a Disconnected DOS Machine

When running DOS disconnected, you must reconnect to the DOS virtual machine to enter the interruption that causes DOS to continue reading jobs. However, you can avoid this annoyance by using the single console image facility. With this facility, you can generate the interruption with the CP SEND command; in this case you do not have to reconnect the DOS virtual machine.

To use CMS to spool jobs to a disconnected DOS virtual machine, spool the reader of the DOS machine with the CONT operand:

```
spool reader cont
```

Then, if another job is received in the card reader of the DOS machine before the currently executing job stream has completed, the job just received is also read and executed. Thus, when running a series of jobs in a single job stream and the reader is spooled with the CONT operand, you can continue to send (from the CMS virtual machine) additional jobs to the DOS virtual machine for execution.

Note: Under these circumstances, you do not have to reconnect to the DOS virtual machine to signal an interruption, unless the currently executing job stream finishes before the next job is received in the card reader.

If DOS is running disconnected and is processing many jobs, you may want to obtain printed output for completed jobs without waiting for all the jobs to finish.

When running VSE with the VSE/Advanced Functions Program Product (5746-XE8), VM/SP automatically releases spooled printer output to VSE/POWER. However, for DOS Release 34 (or earlier), release spooled printer output (either accumulated directly or through the POWER/VS program) by reconnecting to DOS and issuing the command:

```
close printer
```

The 'close printer' command releases the SYSLST output accumulated so far. However, when using the POWER/VS program and a dedicated printer, all spooled output files are under the control of POWER/VS spooling, and not VM/SP spooling. Also, issuing the CLOSE command from a secondary user's console avoids the necessity to reconnect the DOS virtual machine in order to release output.

Console READs and WRITEs in a Disconnected DOS Machine

- *Without the single console image facility:*

When running DOS disconnected, a 15-minute time-out begins when a console read occurs. If the virtual machine does not respond to the read before the 15 minutes elapse, VM/SP automatically logs off the virtual machine.

Example: DOS is running disconnected, and a program running in the machine completes execution or issues a PAUSE request. The virtual machine is logged off after 15 minutes unless you reconnect the virtual machine and issue the BEGIN command.

When running DOS disconnected, VM/SP ignores all output or “writes” to the virtual console unless the console is spooled. To spool virtual console output, you can issue the following CP command before disconnecting the virtual machine:

```
#cp spool console start
```

This command starts recording all console output on a spool file. When you log on again issue:

```
spool console stop close
```

This command stops console spooling and releases the spool file to the real printer. If this command is not entered, VM/SP saves all spooled console output except the last 4K page of output.

- *With the single console image facility:*

If a disconnected virtual machine with an active secondary user issues a read to the console, a message is sent to the console informing the secondary user. No 15-minute time-out is initiated. The secondary user then satisfies the read by issuing a SEND command, containing the appropriate information, to the disconnected virtual machine.

Any console output from a disconnected virtual machine with an active secondary user will appear on the console of the secondary user. Each output line will have a prefix consisting of the disconnected virtual machine’s userid followed by a colon. Also, console spooling can be used as described in “Without the single console image facility”.

Developing and Testing Programs to Run in a DOS Virtual Machine

| The previous discussions noted how to use the System Product Editor and the EXEC facility to help prepare jobs for execution in a DOS virtual machine. In addition to these CMS features, there are a number of other CMS commands for developing and testing programs on CMS disks. One advantage of storing source programs on CMS disks is that they can be maintained as backup copies of a program while a second version is being tested and debugged.

CMS has a special environment, called CMS/DOS, it provides many commands that simulate the functions the DOS environment.

Some of the facilities that are available in CMS/DOS are:

- Creating CMS MACLIBs from DOS system or private Source Statement libraries (edited or unedited members) and assembling programs with these macros directly in CMS using the VM Assembler. (Assembler diagnostic messages are displayed on the terminal. See Appendix B of the *VM/SP Installation Guide* for a sample EXEC used to create a CMS MACLIB from a DOS SSL using the ESERV command.)
- Compiling programs written in DOS COBOL or DOS PL/I programming languages, using DOS macro libraries.
- Displaying or printing the directories of DOS private or system core image, relocatable, source statement, or procedure¹ libraries.
- Displaying or printing the relocatable, source statement, or procedure libraries.
- Link-editing TEXT decks from CMS disks, or relocatable modules from DOS system or private relocatable libraries. These simulated core image libraries called DOSLIBs are on a CMS disk. You can also copy relocatable modules from DOS libraries.
- Loading core image phases from CMS DOSLIBs or from DOS core image libraries into virtual storage and executing them.
- Identifying system and programmer logical units for programs being used and listing current assignments.
- Identifying disk files. When executing programs in CMS/DOS, you can read sequential disk files directly from DOS disks, but cannot write on them. An exception to this rule is when using CMS/VSAM which is capable of reading and writing to VSAM files on DOS formatted disk.
- Using the CMS and VM/SP debugging facilities to debug a program under CMS. These debugging facilities are:
 - CMS DEBUG
 - CP ADSTOP
 - CP TRACE
 - CP PER
 - CP VMDUMP
 - CP DUMP
 - CP TRAP
 - VM/IPCS-E (Program number 5748-SA1)

When a program is tested and debugged in CMS/DOS, you can also prepare a job stream to catalog and execute the program in a DOS virtual machine. For complete details about how to use CMS/DOS, refer to *VM/SP CMS User's Guide*. For details about how to specify CMS commands, refer to *VM/SP CMS Command and Macro Reference*.

¹ VSE/AF private procedure libraries are not supported.

Summary

When a virtual machine user loads DOS into his virtual machine, the terminal becomes the DOS operator console, and the virtual machine user becomes the operator responsible for entering all commands and responses. The three basic techniques for using DOS in a virtual machine are:

1. Running DOS in batch mode
2. Using the IPL command to alternate between DOS and CMS in a single virtual machine
3. Running DOS disconnected with a secondary user.

Before using one of the above techniques, you must understand how to:

- Generate DOS to run in a virtual machine
- Create VM/SP directory entries for DOS virtual machines
- Access the DOS system residence volume
- Ensure that the proper I/O devices are attached to the DOS virtual machine
- IPL and operate DOS under VM/SP

When generating DOS to run in a virtual machine, the primary objectives should be to avoid double CCW translation and to reduce the number of SIO instructions issued by DOS. To meet these objectives, you need to consider how to generate both VM/SP and DOS. (DOS can be generated under VM/SP.)

DOS operation depends upon how DOS was generated. There may be additional operator commands and control statements that must be entered at the console before running jobs on the DOS virtual machine.

There are many ways that DOS virtual machine users can be helped by using the CMS component of VM/SP. The System Product Editor and EXEC facility can be used to help prepare jobs for execution in a DOS virtual machine. CMS commands can be used to develop and test programs on CMS disks. The CMS/DOS environment provides many commands that simulate DOS functions.

Section 4. OS/VS in a Virtual Machine

When loading OS/VS into a virtual machine running under VM/SP, the terminal becomes the OS/VS operator console, and the user is responsible for entering all the commands and responses normally required of the operator.

The four basic techniques to use when running OS/VS in a virtual machine are:

1. **Batch mode:** One user runs as the OS/VS machine (userid OSVS) and other users (like CMSID1) may submit jobs through: the virtual card reader, the system card reader, or via JES remote stations.
2. **Disconnected:** The CMS user who is using the single console image facility acts as the secondary user for the OS/VS user. This allows the CMS user to control messages, replies, and commands for the disconnected OS/VS user at the same physical terminal. The CMS user can perform this function without reconnecting the OS/VS virtual machine.
3. **Disconnected user (OS/VS2 only):** Because OS/VS2 allows a user to recall a prior system message, the user can logon as the OS/VS2 operator under the OSVS userid, start up his system, and then disconnect. While the OS/VS2 machine continues to run, the user can logon as a CMS user (CMSID1) and create and submit job streams and check the resulting output. To check the progress of the operating system, disconnect from the CMS machine and reconnect to the OSVS virtual machine via the LOGON command.
4. **Alternating Technique:** The IPL command is used to alternate between OS/VS and CMS in a single virtual machine. This method requires only one directory entry, (for the OS/VS user). Because of the lengthy IPL process, it is practical only if an installation has created an OS/VS saved system.

Before discussing these four techniques in greater detail, you must understand how to:

- Generate OS/VS to run in a virtual machine
- Create VM/SP directory entries for OS/VS virtual machines
- Access the OS/VS system residence volume
- Ensure that the proper I/O devices are attached to the OS/VS virtual machine
- IPL and operate OS/VS under VM/SP

System Generation Recommendations

When generating OS/VS to run in a virtual machine, you should have these primary objectives:

- To have all commonly used transient routines made resident in storage
- To run all jobs (if possible) as V=R jobs

To meet these objectives, you need to consider how you generate both VM/SP and OS/VS.

For example, OS/VS2 Release 1 (referred to as Single Virtual Storage or SVS) can use two address spaces:

- One address space for V=V jobs (jobs running even though some of their pages are not in real storage)
- Another address space for SVS and any V=R jobs

By defining the SVS virtual machine large enough for all jobs to run V=R, SVS does not have to alternate between address spaces. It should perform better under VM/SP than if it had to alternate between address spaces.

VM/SP Recommendations

When generating VM/SP for an OS/VS virtual machine, note the following recommendations:

VM/SP Saved Systems: IPL time can be reduced by saving any operating system after the generated operating system has been loaded on VM/SP. For more information about generating saved systems, refer to the *VM/SP System Programmer's Guide*.

Handshaking for VS1: VS1 Release 4 and subsequent releases use VM/VS handshaking. VM/VS handshaking permits instructions issued by VS1 in a virtual machine to be processed directly by the processor. It also permits VM/SP to simulate privileged instructions. For further details, refer to the topic "VM/VS Handshaking for VS1" in this section.

IPL Command Enhancement

VM/SP, via the new ATTN parameter of the CP IPL command, can pass a virtual machine console attention interrupt to an OS virtual machine. This activates FASTNIP processing and initializes the OS virtual machine. The ATTN parameter simulates the process of hitting the attention key (or equivalent). If the VS1 virtual machine issues the command in the PROFILE EXEC, an automatic IPL of OS utilizing FASTNIP results without operator intervention. The following process shows how to implement this new function:

1. Set up an AUTOLOG1 virtual machine in the directory. This machine logs on automatically after CP initialization. The AUTOLOG1 directory entry must contain an IPL CMS statement.
2. Issue the CP AUTOLOG command for the VS1 virtual machine in the PROFILE EXEC of the AUTOLOG1 virtual machine.
3. The directory entry of the VS1 virtual machine must contain an IPL CMS statement. The PROFILE EXEC for the VS1 virtual machine specifies the IPL command, with the ATTN parameter, for the VS1 SYSRES volume.
4. OS FASTNIP activates automatically and VS1 initialization completes without operator intervention. The message IEA785I indicates FASTNIP is active and the message IEE048I reveals FASTNIP completion. The VS1 system is ready to accept jobs via the card reader.

Notes:

1. You can manually issue the IPL command with the ATTN parameter or in an EXEC, but you cannot specify it in the directory because the format is invalid.
2. The VS1 environment does not need an attention interrupt to automate the IPL process.
3. If the VS1 virtual machine is autologged, console output messages are lost:
 - Unless you have started console spooling
----- or -----
 - Designated a secondary user to receive the console output from the disconnected VS1 virtual machine
4. If the VS1 virtual machine has a 3270 defined in its directory entry, issue the following CP command prior to the IPLing of VS1. (The command can be entered via the console or in the Profile Exec of the VS1 virtual machine.)

```
terminal conmode 3270
```

OS/VS Recommendations

When generating OS/VS to run in a virtual machine, note the following recommendations:

Specifying Options: Very often options that improve performance on a real machine have no effect (or possibly an adverse effect) in a virtual machine. For example, SEEK separation improves performance on the real machine, but is redundant in a virtual machine. VM/SP itself issues a stand-alone SEEK for all disk I/O.

When VM/SP is the primary operating system and another operating system (such as VS1) is running one or two partitions with a virtual machine under it, generate the other operating system with as few options as possible. This is particularly essential when several virtual machines are sharing the same system residence volume.

When VM/SP is not the primary operating system and the other operating system is being run without VM/SP, generate the other operating system to:

- Be transparent to the users of the other system
- Have the required number of partitions or regions

Sharing the System Residence Volume: Sharing the system residence volume avoids the need to keep multiple copies of the operating system online. The shared system residence volume should be read-only. To share OS/VS among users, remove all data sets with write access from the system residence volume.

OS Recommendations

When generating VS1 to run in a virtual machine, note the following recommendations:

VS1 Storage Limits: The hardware storage size used by VM/SP may be larger or smaller than the hardware storage sizes supported by VS1. In the VM/SP

directory for the VS1 user, use the USER statement to define the minimum and maximum virtual storage sizes for the VS1 virtual machine. For the VS1 system, generate it to support the hardware storage size of the real machine on which the VS1 system is to run.

For example, in the USER directory statement, specify the minimum and maximum storage sizes supported for the VS1 virtual machine. Define the minimum storage size as 1 megabyte -- which is the minimum storage size supported by VS1. Define the maximum storage size according to the VS1 release level (such as VS1 Release 6, that has a 16 megabyte storage limit in nonpaging mode and an 8 megabyte storage limit in paging mode). For the VS1 system to run more efficiently in the VS1 virtual machine, generate its real storage size for 2 megabytes -- the storage size of the real processor. Thus, this system generation specification establishes an initial VM/SP virtual storage size of 2 megabytes even though the minimum VM/SP storage definition was 1 megabyte.

By using VS1 in nonpaging mode, VM/SP handles the paging and CCW translation for VS1, thereby eliminating the VS1 overhead associated with these functions. For a description of VM/VS handshaking between VS1 and VM/SP, refer to the topic "VM/VS Handshaking for VS1" discussed later in this section.

VS1 in a V=R Virtual Machine: When running VS1 in a V=R machine, you can avoid data transfer operations into real page zero by doing one of the following:

- If the VS1 nucleus already exists, replace the existing ORDER statement with the following linkage editor control statement in the VS1 linkage editor deck:

```
ORDER IEAAIH00,IEAIOS00(P)
```

Use all INSERT control statements that were produced during the original VS1 system generation process.

- Generate a new VS1 nucleus prior to stage II execution and perform IOS alignment by modifying the ORDER control statement in the VS1 stage II job stream.

VM/VS Handshaking for VS1

VM/VS handshaking is a communication path between the control program (CP) component of VM/SP and VS1 Release 4 (and subsequent releases) running as a virtual machine under VM/SP.

To improve their operation with VM/SP, systems generated to use VM/VS handshaking can run both in a real machine and in a virtual machine. In a 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 function allows VM/SP to immediately process these output files without operator intervention.
- Providing a nonpaging mode to eliminate duplicate paging.

- Providing a way to avoid a PCI (programmed-controlled interruption) in a BTAM autopoll CCW loop.
- Providing miscellaneous enhancements when running under VM/SP.

Activating VM/VS Handshaking for VS1

Although handshaking is a system generation feature for VS1, it is active only when VS1 is run under the control of VM/SP; it is disabled when that same VS1 operating system is run on a real machine. The VM/VS handshaking feature is active when:

- VS1 is generated with the VM/SP option.
- The virtual machine storage space is at least 1 megabyte.

Note: If nonpaging mode is used, refer to the subtopic “VS1 Nonpaging Mode” described later in this section.

When loading a VS1 virtual machine that has the handshaking feature, the VS1 initialization routines determine whether the handshaking feature is available. VS1 checks for VM/SP by issuing an STIDP (store processor ID) instruction. When STIDP returns the version code X'FF', VS1 is running under VM/SP.

On receiving version code X'FF', VS1 issues a DIAGNOSE code X'00' instruction to store the VM/SP extended-identification code. If VM/SP returns a code to VS1, VM/SP supports handshaking; otherwise, VM/SP does not support handshaking.

VS1 Nonpaging Mode

When both VM/SP and VS1 support handshaking, full handshaking results when VS1 runs in nonpaging mode. However, handshaking does not require nonpaging mode. When VS1 is run under the control of VM/SP, it executes in nonpaging mode if:

- Its virtual storage space is equal to the storage space of the VM/SP virtual machine.
- Its virtual machine storage space is at least 1 megabyte.
- VM/VS handshaking is available.

Providing the above conditions are satisfied when VS1 is loaded under control of VM/SP, VS1 issues a special message after the IEA760A SPECIFY VIRTUAL STORAGE SIZE message:

```
IEA788I NON-PAGING MODE OF VS UNDER VM/SP
```

When VS1 executes in nonpaging mode, it uses fewer privileged instructions and avoids duplicate paging. The VS1 nucleus initialization program (NIP) fixes all VS1 pages to avoid duplicate paging.

Note: The VM/SP working set size (the estimated number of real storage pages that a virtual machine needs to execute) may be larger for a VS1 virtual machine in nonpaging mode than for one in paging mode.

Considerations unique to nonpaging mode are:

- ***Responding to Virtual storage size message***

An EOB or U response to the VS1 SPECIFY VIRTUAL STORAGE SIZE message automatically sets the VM/SP virtual storage space equal to the VS1 virtual machine storage space, provided the latter is 1 megabyte or larger.

- ***Nonpaging mode storage limits***

Storage limits for nonpaging mode are the same as for VS1 itself:

- VS1 Release 6 has a 16 megabyte storage limit in nonpaging mode and an 8 megabyte limit in paging mode (the limit is equal to the real machine size or configuration as supported by VS1). During IPL, VS1 issues a message that specifies these limits.
- VS1 Release 5 has a 4 megabyte storage limit and issues a message to that effect.
- VS1 Release 4 has a 4 megabyte storage limit, but issues no message to that effect.

- ***Providing a minimum size VS1 nucleus***

A minimum size VS1 nucleus tends to be more suitable for the nonpaging mode. It provides more space for problem program partitions.

- ***VS1 storage mapping unaffected by nonpaging mode***

In nonpaging mode, VS1 maps virtual storage normally; that is, it puts these functions in the high addresses of virtual storage: the JES buffer pool, VTAM workspace, RTAM area, JES routines, resident modules, and the pageable supervisor. By minimizing the virtual storage requirements for these functions, VS1 can provide more problem program partition space.

- ***VS1 paging data set not used in nonpaging mode***

The VS1 paging data set is not used in nonpaging mode. The VS1 system does not require page parameters and page packs.

Note: If a VS1 standalone dump should be taken, do not dump VS1 page data sets at MSGHMD021A. If an attempt is made to dump page data sets, a VM/SP abend PTR007 may occur.

- ***Executing V=R jobs in nonpaging mode***

It is possible to execute V=R jobs in nonpaging mode. The V=R line is a valid IPL parameter and the V=R logic within VS1 is applicable. However, any benefits from such operation would appear to be questionable because the entire VS1 system functions V=R in nonpaging mode.

- ***FASTNIP forcing of nonpaging mode***

FASTNIP automatically forces nonpaging mode when the virtual machine storage space is 1 megabyte or larger.

Note: There may be occasions to run a VS1 system in a virtual machine with a storage space of 1 megabyte or larger and not use nonpaging mode. To avoid initiating nonpaging mode, specify the virtual storage space explicitly to VS1 in response to message IEA760A; for example, R00, '6144'.

The following examples show whether nonpaging mode is initiated when initializing VS1 in a VM/SP environment with handshaking.

Example 1:

```
VM/SP Size = 768K
OS Virtual Storage Size = 6M
IEA760A Response = 'EOB'
```

Nonpaging mode is not initiated, because the virtual machine storage space is less than 1M.

Example 2:

```
VM/SP Size = 1M
OS Virtual Storage Size = 6M
IEA760A Response = 'U'
```

Nonpaging mode is initiated, and VS1 forces the virtual machine storage space to 1 megabyte. The VS1 system may not complete initialization because of insufficient virtual storage space.

Example 3:

```
VM/SP Size = 4M
OS Virtual Storage Size = 4M
IEA760A Response = R00, '4096'
```

Nonpaging mode is initiated, because the explicit response happens to equal virtual machine storage space.

Example 4:

```
VM/SP Size = 3M
IEA760A Response = R00, '6144'
```

Nonpaging mode is not initiated. The explicit response sets virtual storage space to 6 megabytes, which VS1 requires for paging space.

Closing CP Spool Files

When VM/VS handshaking is active, VS1 closes the CP spool files when the job output from the VS1 DSO, terminator, and output writer is complete. Once the spool files are closed, VM/SP processes them and sends them to the real printer or punch without operator intervention.

During its job output termination processing, VS1 issues DIAGNOSE code X'08' instructions to pass the CP CLOSE command to VM/SP for each CP spool file.

VS1 Without Handshaking

When a nonhandshaking copy of VS1 (prior to Release 4.0) is run under VM/SP, the storage considerations are the same as if VS1 were running in native mode. The VS1 virtual storage space must be at least 512K larger than the virtual machine storage space.

The VS1 virtual storage space is specified at VS1 system generation time and can be altered in response to the IPL message IEA760A SPECIFY VIRTUAL STORAGE SIZE. The virtual machine storage space is specified initially in the VM/SP directory and can be altered by using the CP DEFINE STORAGE command.

BTAM Autopoll CCW Change Detection

When an operating system in a virtual machine has enabled the BTAM autopoll feature, it notifies VM/SP (via a DIAGNOSE instruction) whenever the autopoll CCWs are modified. VM/SP then modifies the real CCWs and does not check the autopoll CCWs for modifications each time the string is executed. This CCW change detection reduces VM/SP overhead and thereby improves the overall performance.

Note: If the autopoll feature is disabled for VM/SP (by the SET AUTOPOLL OFF command), a performance degradation occurs.

Miscellaneous Enhancements

When VS1 without handshaking is run in the VM/SP environment some duplication of function results. Because VS1 must perform certain functions when it is run on a real machine, it continues to perform all those functions in a VM/SP virtual machine, even though VM/SP also provides the services. However, with handshaking, VS1 avoids using many instructions and procedures that are redundant or less efficient in the VM/SP environment.

In either paging or nonpaging mode, VS1 avoids using:

- ISK (insert storage key) and SSK (set storage key) instructions; instead, VS1 uses a protection key table
- Seek separation for 2314 direct access devices
- The ENABLE/DISABLE sequence in the VS1 I/O supervisor (IOS)
- TCH (test channel) instructions preceding SIO instructions
- PCI (program-controlled interruptions) in the BTAM autopoll CCW sequence

In nonpaging mode, VS1 with handshaking avoids using:

- LRA (load real address) and RRB (reset reference bit) instructions (this is especially important when virtual machine assist is not enabled)
- The DIAGNOSE code X'10' instruction to release virtual pages or discontinuous storage
- VS1 paging and CCW translation

IBM 3850 Mass Storage System Considerations

There are no special system generation requirements when generating OS/VS (OS or OS) to use the MSS and operate in a virtual machine. Any VS1 or MVS system that supports the MSS can use VM/SP MSS support. VM/SP MSS support allows a VS1 or MVS virtual machine to:

- Use a dedicated mass storage control (MSC) port (or channel interface) and dedicated 3330V devices

----- and -----

- Act as the host for the VM/SP communicator program (which communicates 3330V mount and demount orders and responses between the MSC and VM/SP)

However, this support requires each 3330V address defined in OS/VS to be identical to the 3330V address defined both to VM/SP and to the virtual machine using it.

For details about how to generate VM/SP to support the MSS and to install the VM/SP communicator program in either VS1 or MVS, refer to the *VM/SP Planning Guide and Reference*. For details about how VM/SP communicates with the MSS and uses it as well as how to provide backup and recovery for MSS volumes, refer to the *VM/SP System Programmer's Guide*. For details about MSS initialization, refer to the *VM/SP Operator's Guide*.

Sample OS/VS Directory Entries

The following directory entries represent some batch type virtual machines that can be used to run production jobs under OS and OS/VS. The operands specified on the OPTION control statements reflect the requirements of the particular system being used. Disk space can either be dedicated or shared with other systems.

An MFT Virtual Machine:

```
| USER OSMFT PASSWORD 1M 1M G
| ACCOUNT ACCTNO BIN5
| IPL 230
| OPTION REALTIMER ISAM BMX
|     CONSOLE 01F 3215
|     SPOOL 00C 2540 R
|     SPOOL 00D 2540 P
|     SPOOL 00E 1403
|     DEDICATE 230 OSRES
|     DEDICATE 231 OSWRK
|     DEDICATE 185 285
|     DEDICATE 186 286
```

A VSI Virtual Machine:

```
| USER OSVS1A PASSWORD 1M 1M G
| ACCOUNT ACCTNO BIN6
| IPL 350
| OPTION REALTIMER VIRT=REAL ECMODE BMX
|     CONSOLE 01F 3215
|     SPOOL 00C 2540 R
|     SPOOL 00D 2540 P
|     SPOOL 00E 1403
|     SPOOL 012 3505
|     SPOOL 002 3211
|     LINK MAINT 190 190 RR
|     LINK MAINT 19D 19D RR
|     LINK MAINT 19E 19E RR
|     MDISK 191 3330 21 10 UDISKA WR RPASS WPASS
|     MDISK 350 3330 0 100 VOSDOS MW
|     MDISK 351 3330 51 50 UDISK1 W
```

Another VSI Virtual Machine:

```
| USER OSVS1B PASSWORD 1M 1M G
| ACCOUNT ACCTNO BIN7
| IPL 350
| OPTION REALTIMER ECMODE ISAM BMX
|     CONSOLE 01F 3215
|     SPOOL 00C 2540 R
|     SPOOL 00D 2540 P
|     SPOOL 002 3211
|     LINK MAINT 190 190 RR
|     MDISK 191 3330 31 10 UDISKA WR RPASS WPASS
|     MDISK 350 3330 100 100 VOSDOS W
|     MDISK 351 3330 0 50 UDISK3 W
```

An MVS Virtual Machine: (for running test jobs only)

```
| USER MVSSP PASSWORD 4M 16M BCG
| ACCOUNT ACCTNO BIN8
| IPL CMS
| OPTION REALTIMER ECMODE BMX 370E
|     CONSOLE 01F 1052
|     SPOOL 01C 2540 READ A
|     SPOOL 01D 2540 PUNC A
|     SPOOL 007 3211 A
|     SPOOL 008 3211 A
|     SPOOL 017 3211 A
|     SPOOL 018 3211 A
|     SPOOL 01E 1403 A
|     SPECIAL 2FF 3270
|     SPECIAL OFF TIMER
|     LINK MAINT 190 190 RR
|     LINK MAINT 19D 19D RR
|     LINK MAINT 19E 19E RR
```

(This entry then uses the MVS machine's CMS PROFILE EXEC to attach the appropriate volumes and IPL MVS.)

Accessing OS/VS

This topic assumes that OS/VS for use under VM/SP has already been generated and that the system residence volume is available on a real disk or minidisk in read/write status.

As the OS/VS operator, the OS/VS user needs to know the location of the system residence volume. Its location can be defined in the virtual machine configuration in one of three ways:

1. Define the system residence volume as a read/write disk in the directory entry for the OSVS userid. Such a definition may appear as follows:

```
MDISK 250 3330 0 404 VS2RES WR YOUR NAME
```

Many installations prefer this approach for maintaining a directory definition of the system residence volume.

2. Use the CP LINK command to define the system residence volume after logon. For example, if the SVS or MVS system programmer “owns” the system residence volume and keeps it in his virtual machine at virtual address 150, the OSVS user could gain access to it with this CP command:

```
link vs2sysp 150 250 w name1
```

In this command VS2SYSP is the programmer’s userid, and NAME is the write password.

3. Have the VM/SP system operator (or any class B user) exclusively attach the entire system residence volume to the OSVS userid by issuing this command:

```
attach 152 to osvs as 250
```

In this command, 152 is the real device address on which the system residence volume is mounted.

Using Virtual Devices

When using OS/VVS in a virtual machine, the user is the OS/VVS operator. This user must have the following devices, that are normally defined in the VM/SP directory entry:

- A virtual card reader, from which OS/VVS reads the OS/VVS input job stream.
- A virtual printer, that handles the printed output generated by OS/VVS.
- The virtual punch receives punched output generated during OS/VVS operation.

In addition to these unit record devices, the OS/VVS operator can attach virtual tape and direct access storage devices to the virtual machine (by using either the ATTACH or DEFINE commands). The user can also specify these devices in the VM/SP directory entry.

Depending upon how OS/VVS was generated, you may need to change a virtual device address. For example, if OS/VVS expects a 3211 printer at device address 002 and the directory entry does not contain this assignment, define one with the CP DEFINE command:

```
define 3211 002
```

Before using OS/VVS, find out from the OS/VVS system programmer what are the installation’s virtual device requirements.

¹ If an installation is using password-on-the-command-line suppression, a user cannot specify the password on the same command line. Passwords must be entered in such a way that they are either not displayed on display terminals or typed upon a mask for typewriter terminals.

Defining the Operator's Console

The operator's console must be at the address specified during OS/VS system generation. The easiest way to ensure this is to define the appropriate console address in the directory entry. For example, the CONSOLE directory control statement could appear as follows:

```
CONSOLE 01F 3210
```

This statement defines a virtual 3210 console at virtual address 01F.

Using the VM/SP Spool File System

You should let the VM/SP spool file system handle printer or punch output that does not have to be printed or punched. For example, when using the alternating technique, route print output to the virtual card reader by using this CP SPOOL command:

```
#cp spool printer to *
```

After issuing this command, you can subsequently load the CMS system and create a CMS file from the data in the virtual reader (by using the CMS READCARD command). Then, by using the System Product Editor, you can scan the contents of this data at your terminal.

Preparing Jobs for an OS/VS Virtual Machine

Prepare and submit a job stream to an OS/VS virtual machine in one of two ways:

1. Place a deck of real punched cards that contain the appropriate job control, program and data in the real card reader. Place a CP ID statement at the beginning of this job stream deck to indicate the OS/VS userid. For example:

```
ID OSVS
```

```
----- or -----
```

```
USERID OSVS
```

Either statement is a valid ID statement for directing the input that follows to the OSVS user's virtual reader (the reader with the lowest virtual device address).

2. Use the CMS system to create a CMS file containing images of what would normally be submitted through a card reader on a real System/370. Enter the CP SPOOL command to cause subsequent punched output to be directed to the virtual card reader of the OS/VS machine. Enter the CMS PUNCH command to generate the virtual card deck:

```
cp spool punch to osvs  
punch vsjob27 jcl (noheader)
```

The NOHEADER option of the PUNCH command suppresses punching a CMS READ control card at the beginning of the deck.

A job stream spooled to OS/VS by either of these methods remains in the card reader of the OS/VS virtual machine until you start an OS/VS reader.

When spooling jobs to a virtual machine, clear any data that may remain in the virtual punch from previous jobs by issuing these CP commands:

```
cp spool punch nocont
cp close punch purge
```

These commands ensure that the virtual punch is purged of any existing reader files. The first command is required if the punch had been originally spooled with the CONT operand.

Job Entry and Output Retrieval

When running OS/VS in a virtual machine, a primary consideration is job entry and output retrieval. Several techniques can be used to achieve these functions:

- Use the OS/VS virtual machine in batch mode where it operates as OS/VS in native mode. It reads in job streams through a dedicated card reader and prints output generated by the virtual machine on a dedicated printer.
- A single directory entry (userid) can contain a configuration sufficient for running both CMS and OS/VS (the alternating technique). Load CMS to create and edit OS/VS job streams and to check the OS/VS output. Load OS/VS to run the OS/VS job streams.
- Use two different userids to keep two virtual machines running. (This is the optimum environment.) Use one userid for the OS/VS machine while using the other for a CMS machine to create job streams and inspect output. By using the CP DISCONN command, you can run both virtual machines from the same terminal. Thus, both machines are running concurrently, but you communicate with only one at a time. If two terminals are available, each system can run independently of the other.

Loading OS/VS

At logon use whatever userid has been set aside exclusively for the OS/VS machine or your own userid that you use when running alternating or concurrent systems.

```
logon osvs
```

Because extended control mode (EC) is required to operate OS/VS, the directory entry should contain the ECMODE specification in the OPTION control statement. If it is not included, enter this CP command to enable extended control mode simulation:

```
set ecmode on
```

Note: To run OS you do not need the ECMODE option, unless it is generated for a System/370. You also do not need the ECMODE option unless you are running the generalized trace facility (GTF) under OS. However, OS normally requires the REALTIMER option.

At this point, between logging on and loading the operating system, you may find it desirable or necessary to alter the virtual machine's storage size. For example, if the directory entry specifies 1 megabyte of storage and you need 2 megabytes for a particular terminal session, issue the CP DEFINE command:

```
define storage as 2m
```

Virtual machine storage can be redefined up to the limit set in the USER control statement of the directory entry.

If OS/VIS is generated with a console address for display mode, then you can change your console address (via the DEFINE command) to the display mode address to make use of the full screen OS/VIS operator's console. You must change the CONMODE to 3270 via the TERMINAL command. Assuming the console is at 01F and OS/VIS has a display mode console at 007, then issue the following commands:

```
define 01F 007
terminal conmode 3270
```

If the guest operating system is OS/VIS2 MVS, one more terminal command is required for display mode to function properly. The "screen save" option causes CP to save the MVS console screen image so that transitions from CP mode to full screen mode and back can be made. The following command will enable this facility:

```
terminal scrnsave on
```

CP commands can be issued only from CP mode. If you are in CONMODE 3270 you can do this by pressing the PA1 key or its equivalent. To return to full screen mode, issue the BEGIN command or press the PA1 key.

Once the IPL volume is made available, load OS/VIS by entering this command:

```
ipl 250
```

OS/VIS responds with a "SPECIFY SYSTEM PARAMETERS" message. The proper use of a system parameter list (IEASYS00 or IEASYSxx), created during or subsequent to the OS/VIS system generation process, can result in a significant saving of time. Commonly used operator commands can be placed in the SYS1.PARMLIB data set to shorten the IPL process.

OS/VIS Operation

To control OS/VIS, use OS/VIS operator commands to hold and release queues and jobs and to start initiators or define partitions. Users can observe the progress of the command's execution by following the OS/VIS messages. Figure 4-1 on page 4-15 shows how VM/SP loads VS1 in a virtual machine.

LOGON AT 01:96:14 EST WEDNESDAY 03/09/83

def stor 3072k

STORAGE = 03072K

q set

MSG ON , WNG ON , EMSG TEXT , ACNT ON , RUN OFF
LINEDIT ON , TIMER REAL , ISAM OFF , ECMODE ON
ASSIST ON SVC , PAGEX OFF , AUTOPOLL OFF
IMSG ON , AFFINITY OFF

def 009 01f

CONS 01F DEFINED

q chan

CHANNELS = SEL

def chan bmx

CHANNELS = BMX

The virtual selector channels have been redefined to block multiplexer. This is a performance consideration to improve the processing of I/O operations. To avoid issuing the CP DEFINE BMX command, the VM/SP directory entry can specify the BMX option in the OPTION control statement.

i 250

IEA760A SPECIFY VIRTUAL STORAGE SIZE

r 00,'3072'

Specifying the size of the virtual machine's storage relative to the VS1 virtual storage size is a performance consideration. VS1 has been shown (in performance studies) to operate more efficiently in a virtual environment if it is not forced to do any of its own paging. Also, in nonpaging mode VS1 avoids many privileged instructions, thereby reducing VM/SP overhead.

To force VS1 into a nonpaging mode, use VM/VS handshaking and define the storage size for the virtual machine equal to the virtual storage size requested by VS1.

IEA788I NON-PAGING MODE OF VS UNDER VM/SP

IEE054I DATE=75.300,CLOCK=009.00.00

IEE054I DATE=75.300,CLOCK=009.00.02,GMT

IEA764I NIPNULL,CMDMON,DFNPARM,JESNULL,,,,,SETPARM,,

IEA101A SPECIFY SYSTEM AND/OR SET PARAMETERS FOR RELEASE 04.0 OS

r 00,'u'

IEA106I IEAAPFOO NOT FOUND IN SYS1.PARMLIB

IEE140I SYSTEM CONSOLES

CONSOLE/ALT	COMD	AUTH	ID	ROUTCP
01F/01F	M	ALL	01	1-10,10-16

IEF031I SYSGEN VALUES TAKEN FOR JES

IEF866I DEFINE COMMAND BEING PROCESSED

IEE804I PO=(C=AOB,576'K,A,I),P1=(C=AOB,576K,A,E,LAST),

IEE804I P2=(INACTIVE),P3=(INACTIVE),

IEE804I P4=(INACTIVE),P5=(INACTIVE),

IEE804I P6=(INACTIVE),P7=(INACTIVE),

IEE543I 250K BYTES FREE SPACE

IEE817I TMSL=NONE

IEE805I DEFINITION COMPLETED

IEE101A READY

IEE009I JLPRM=(U)

IEE050I MN JOB NAMES,T

IEE048I INITIALIZATION COMPLETED -

Figure 4-1. Sample IPL of VS1 under VM/SP

When using CMS, initial operator commands can be incorporated into a CMS EXEC procedure as part of the job stream. For example, create the the following EXEC procedure called SETUPVS, to issue these commands:

```
CP LINK VSSYS 250 250 RR OSPASS
CP DEFINE 009 AS 01F
CP IPL 250
```

After starting the appropriate OS/VS readers, the virtual machine is ready to receive input from card readers, DASD, or tape drives.

Communicating with CP

During OS/VS virtual machine operation, you can issue CP commands to: (1) communicate with the VM/SP system operator or other virtual machine users, and (2) query and alter the status of the configuration and spool files. In general, you can enter any of the CP commands permitted under your userid's privilege class.

Entering CP commands while an OS/VS virtual machine is running depends on the terminal mode (as defined by the CP `TERMINAL` command or its default value). When not running as the VM/SP system operator, the default terminal mode is VM. In this mode, pressing the attention key once (or its equivalent) passes an interruption pending condition to the OS/VS virtual machine. Pressing the attention key twice places the virtual machine in the CP command environment from which CP commands can be entered. For a complete description about how to use the attention key, refer to the *VM/SP CP Command Reference for General Users*.

Using the #CP Function

In most cases during virtual machine operation, you can use the #CP function to enter CP commands directly from the virtual machine. If the virtual machine has issued a read to the terminal, enter a CP command with the #CP function. For example, when no longer using virtual tape drive 397 mapped to real tape drive 492, issue:

```
#cp detach 397
#cp msg op I am done with real drive 492
```

VM/SP immediately processes these command lines, and the virtual machine read remains outstanding.

Note: You may not always be able to enter CP commands with the #CP function. The read issued by OS/VS at the terminal must be for at least as many bytes as entered in the #CP command line; any additional information is truncated. If the read is at least three bytes, enter the #CP command. This command places the user in CP command mode from where CP commands can be entered directly. To return to the virtual machine environment, enter the `BEGIN` command.

When in `CONMODE 3270` '#CP' will not be intercepted by CP. You must use the PA1 key or its equivalent. See "Loading OS/VS" discussed earlier in this section.

Using OS/VS in Batch Mode Under VM/SP

When many users submit jobs to a single OS/VS virtual machine, someone is generally needed to tend the machine as an operator. The virtual machine operator must make those decisions required of an operator on a real machine; that is, deciding what work is going to be done and what is the most efficient way of doing it.

In batch mode, one user runs as the OS/VS machine (userid `OSVS`) and other users (like `CMSID1`) may submit jobs either through the virtual card reader,

through the system card reader, or through JES remote stations. If the card reader is not dedicated to the OS/VS virtual machine, place an ID card at the beginning of each job stream, such as:

```
USERID OSVS A
```

The USERID (or ID) indicates the valid beginning of an ID card, OSVS is the name of the VM/SP user to receive the card input in his virtual reader, and A is the VM/SP reader class.

Note: If the VM/SP system operator has dedicated a card reader to the OS/VS virtual machine, then the ID card must be omitted at the beginning of the deck.

You can send jobs to the OS/VS machine from other virtual machines by spooling your punch to the OS/VS userid, such as:

```
#cp spool punch to osvs
```

Entering this statement causes subsequent punched output to appear in the virtual card reader for userid OSVS.

Alternating Between CMS and OS/VS Under VM/SP

When working in a program development environment (rather than a production environment) and unable to test programs directly under CMS, you can alternate between OS/VS and CMS in a single virtual machine. Some advantages to this technique are:

- Reduced unit record output. Users can examine program output and compiler listings online, check the results, and resubmit the job without producing any output on the system unit record devices.
- Faster turnaround time (generally) than in a batch environment.

Before using this technique, you should be familiar with the System Product Editor. CMS file manipulation commands can be found in the *VM/SP CMS User's Guide*.

Loading CMS into a Virtual Machine

To load CMS into a virtual machine, use the CP IPL command and specify either a saved system name or a device address:

```
ipl cms
```

```
----- or -----
```

```
ipl 190
```

When CMS responds with a message like this:

```
CMS VM/SP 3.0
```

enter the CMS commands to create an OS/VS job stream.

Using the CMS Editor To Prepare Job Streams

The following CMS procedure creates an OS/VS job stream that can be passed to the OS/VS virtual machine's reader. It shows how to compile a PL/I program under OS/VS, making the PL/I source file available as a CMS file called PLI27 DECK.

Commands	Explanations
xedit pli127 jcl	open a CMS file by name
input	go into input mode
//pli127 job cps,fred,msglevel=1	enter jcl entries
//cat exec plifc	.
//sysin dd *	.
(null line)	return to edit mode
getfile pli127 deck	copy over PL/I source file
input	return to input mode
/*	enter jcl entries
//	.
(null line)	.
file	return to edit mode and write the file to disk

Issuing Spool Commands to Control Unit Record Devices

Spool the virtual punch to the virtual machine.

```
cp spool punch to *
```

This command causes subsequent punched output to appear in your own virtual card reader. To submit a job to the OS/VS machine, punch the JCL and associated card data.

Spool the virtual printer to the virtual card reader.

```
cp spool printer to *
```

Instead of routing printed output to the real printer, this command causes it to appear in the virtual card reader. By using CMS, each print file can then be read, examined, and either purged or printed at the real printer.

Note: Since you may find both punch and printer files in the virtual reader, consider using the spool file class attribute to control which files are to be processed at any one time.

Punching CMS Files

Use the CMS PUNCH command to transfer the job stream to the virtual card reader of the OS/VS virtual machine.

```
punch pli27 job (noheader
```

By specifying NOHEADER option of the PUNCH command, VM/SP does not punch a CMS READ control card at the beginning of the output deck.

Initializing OS/VS

Load OS/VS into the virtual machine by issuing this command:

```
cp ipl 250
```

When an OS/VS reader is started, it reads the job stream that had been previously punched with the punch spooled to your own userid. For example, the OS/VS command:

```
s rdr,00c
```

starts a reader on virtual device 00C and reads those cards that appear in the OS/VS reader queue.

Reloading CMS into a Virtual Machine

When the job stream has been processed, reload CMS and use the READCARD command to create a CMS file from the printed output and put it onto a CMS disk. The output can now be examined with the System Product Editor or TYPE command. When hardcopy output is needed, the file can be printed via the CMS PRINT command. When using the READCARD command to create a CMS file, do not use the filetype of LISTING. If you do, VM/SP assumes the first character of each line to be a control character and it is not printed.

Examining OS/VS Virtual Machine Output

To examine a job's output executed under the OS/VS virtual machine, spool the unit record output to the your own userid. Now, read the file in the virtual card reader onto a CMS disk by using CMS READCARD command:

```
readcard pli27 job
```

The XEDIT command can be used to scan the file.

If programming errors occurred during the execution of the job stream, the System Product Editor can be used to make corrections to the source program. Using the System Product Editor you can also correct job control statements, or resolve any other execution problems. After making these corrections you can resubmit the job.

Dynamic SCP Transition to or from Native Mode

Prior to VM/SP support of dynamic System Control Program (SCP), transition to or from native mode was inconvenient. Installations would find it troublesome to transfer control of an operating system from VM/SP virtual machine mode to native mode or vice versa. They had to shutdown their operating system and IPL it again. With the transition function, you can make such a transition without shutting down the operating system and initializing it again. Thus, you avoid the overhead associated with continuously running an operating system under VM/SP, but can still have VM/SP function when it is needed.

VM/SP support of dynamic SCP transition is primarily for SVS and MVS operating systems. However, the VS1 operating system running without VM/VS handshaking can also use this support. This support is not provided for operating systems that run in BC mode.

VM/SP makes the transition from VM/SP to native mode (or vice versa) as transparent to the operating system user as possible. Before making this transition, the operating system must run in a V=R virtual machine with dedicated I/O devices. The V=R mode is necessary because the storage occupied by the operating system under VM/SP and the storage used in native mode *must* have the same addresses. The transition function requires dedicated devices because the I/O devices must have the same addresses while running under VM/SP (virtual I/O addresses) as when running without VM/SP (real I/O addresses).

Restriction for MVS Virtual Machines

The transition function does not support either real or virtual channel reconfiguration. Thus, for an MVS system to operate in a virtual machine and make a dynamic transition to native mode, do not generate the system with channel reconfiguration hardware (CRH) support. That is, do not specify `OPTIONS=(CRH)` in the MVS CTRLPROG system generation macro.

Operating Procedures

To transfer control of an operating system from under VM/SP to native mode, follow these steps:

1. All VM/SP users must logoff, except for the VM/SP system operator and the V=R virtual machine.
2. The V=R virtual machine user must:
 - a. Insure that the virtual device addresses of the dedicated devices match the same real I/O addresses.
 - b. Insure that there are no minidisks.
 - c. Stop all virtual spooling devices in the virtual machine.
 - d. Issue the CP CLOSE command to close any open spool files.
 - e. Issue the CP DETACH command to detach these virtual spooling devices.
 - f. Ensure that the interval timer is turned off (by using the CP SET TIMER OFF command) if it is not used by the virtual machine.
 - g. Ensure that no activities are being traced and no address stops are being set.
3. The VM/SP system operator must drain the real unit record devices.

Note: The transition to native mode cannot take place as long as any outstanding I/O events exist for dedicated devices that belong to the MVS virtual machine. This includes teleprocessing lines.

When these conditions are met and VM/SP is running in uniprocessor mode, the VM/SP system operator can issue the CP class A command `QVM userid` to give the V=R virtual machine control of the processor. However, if an installation doesn't want to return from native mode to VM/SP, they can have the VM/SP system operator issue the `QVM userid` command with the `NORETURN` operand.

Note: When the operating system is in native mode, the operating system user:

- Has normal use of the restart PSW key. It can be used to present a restart interruption to the native operating system.
- Must not alter the storage above the VM/SP V=R storage area. This storage contains the VM/SP nucleus, which VM/SP requires to transfer control of an operating system from native mode back to VM/SP.

Before transferring control of an operating system from native mode back to VM/SP virtual machine mode, the operating system user must ensure that the I/O devices and their addresses are identical to those previously used under VM/SP. Once these conditions are met, the VM/SP system operator must follow these steps at the system console to transfer operating system control from native mode to VM/SP virtual machine mode:

1. Display the restart PSW at main storage location zero.
2. Use steps 2a, 2b, and 2c to determine whether the instruction address in the PSW is for VM/SP or the native operating system. (The instruction address is in the last three bytes of the PSW.)
 - a. Check the translation bit in the restart PSW. If it is on, change it to off. The operating system may have refreshed the restart PSW and turned on the translation bit. The translation bit must be turned off when transferring control to VM/SP so that the hardware will not attempt to translate the real address.
 - b. For VM/SP, if the address points to either entry point DMKQVMRS or DMKQVMRX in module DMKQVM, the operator can proceed to step 3. (The entry point addresses are in the CP load map produced either by the system generation process or by the installation whenever it changes the CP nucleus.)
 - c. For the native operating system, the address points to the system's restart interrupt handler when the system has done a PSW refresh. The VM/SP system operator must change this address to either entry point DMKQVMRS or DMKQVMRX (as listed in the CP load map described for step 2b). Enter the address for DMKQVMRS when the operating system does not use the OS/System Extensions Program Product, 5740-XE1. Otherwise, enter the address for DMKQVMRX.
3. Subtract eight bytes from the instruction address in this PSW.
4. Locate the storage area pointed to by the address determined in step 3 and store X'FF' in the first byte of that location. Do not change the other three bytes.
5. Press the restart key to return control to VM/SP.
6. Since the system operator is not autologged back on, it is necessary to logon the operator explicitly from the VM Console.

Notes:

1. If VM/SP was generated to support an AP system, VM/SP can resume AP operations in the attached processor when the system resource operator (class B) varies it online.
2. When running MVS with TCAM active in a virtual machine, and dynamic translation is derived for MVS, TCAM should be generated with the VM=YES option specified. The VM "SET NOTRANS ON" command should always be used for the V=R MVS virtual machine.

Error Recording

When an SCP makes the transition to native mode, error records for the SCP are in two locations:

1. When under VM/SP, VM/SP records them in its error recording cylinders.
----- and -----
2. When in native mode, the SCP records them in its SYS1.LOGREC data set.

To find all the error records that pertain to the SCP, you must look in both locations. To put the records in chronological sequence, you can follow the time and date recorded in each record.

Using More Than One Virtual Machine

You can run multiple systems, each in its own virtual machine and each controlled from its own terminal. However, if multiple terminals are not available, and the Single Console Image Facility is not used, you can use one terminal to control all systems, but only one at any one time.

This approach combines the alternating system and batch techniques. Separate userids are used to run OS/VVS in one virtual machine and to prepare jobs and examine OS/VVS output in a CMS virtual machine.

After submitting a job stream under the CMS userid, issue the DISCONN or LOGOFF command (depending upon how soon you intend to logon to the CMS ID again). The terminal is now free to be used for running the OS/VVS job stream under the OSVS userid:

```
logon cmsid
.
.
.
(route jobs to OSVS user)
.
.
.
disconn
logon osvs
.
.
.
(run jobs)
```

The procedures for running with two virtual machines are much the same as those used in running a single virtual machine in alternating mode. The primary difference is that you now spool the virtual punch and printer to the other virtual machine instead of spooling them to your own userid:

```
logon cmsid
sp pun osvs
.
.
(route jobs to OSVS user)
.
.
disconn
logon osvs
#cp sp prt cmsid
#cp sp pun cmsid
.
(run OS/VS jobs)
.
.
```

Disconnection Considerations

When using more than one userid to alternate communications between operating systems, consider:

- How OS/VS may read additional jobs from the card reader.
- What happens when a read is issued at the disconnected OS/VS virtual console.
- What happens to the console output of the disconnected virtual machine.
- If you are using the single console image facility to control both virtual machines concurrently from one terminal.

Sending Jobs to a Disconnected OS/VS Machine

When using CMS to route jobs to a disconnected OS/VS virtual machine, spool the OS/VS reader with the CONT operand of the CP SPOOL command:

```
spool reader cont
```

This command allows the OS/VS reader to read more than one job at a time without operator intervention.

Console READs and WRITEs in a Disconnected OS/VS Machine

- ***Without the single console image facility:***

When running OS/VS disconnected, a 15-minute time-out begins when a console read occurs. If the virtual machine does not respond to the read before the 15 minutes elapse, VM/SP automatically logs off the virtual machine.

Whenever running OS/VS disconnected, it is suggested that a console log be created. This log provides you with a history of what jobs were run. It can also indicate any unusual circumstances that occurred during the terminal session.

To start console spooling, issue:

```
spool cons start
```

To stop console spooling and to print the log, issue:

```
spool console stop close
```

When a virtual machine is running disconnected, all console output is lost unless you initiate console spooling, by issuing:

```
#cp spool console start
```

Spooling of the console output continues until you either log off or issue:

```
#cp spool console stop
```

Disconnecting the virtual machine does not stop console spooling. Therefore, the spooled console log for a terminal session, punctuated with several disconnects, consists of one uninterrupted printer file.

- ***With the single console image facility:***

If a disconnected virtual machine with an active secondary user issues a read to the console, a message is sent to the console informing the secondary user. No 15-minute time-out is initiated. The secondary user then satisfies the read by issuing a SEND command to the disconnected virtual machine.

Any console output from a disconnected virtual machine with an active secondary user will appear on the console of the secondary user. Each output line will have a prefix consisting of the disconnected virtual machine's userid followed by a colon. Also, console spooling can be used as described in "Without the Single Console Image Facility."

Developing and Testing Programs to Run in an OS/VS Virtual Machine

The previous discussions demonstrated how the System Product Editor and EXEC facility can help a user's prepare jobs for execution in an OS/VS virtual machine. In addition to these CMS functions, there are a number of other CMS commands for developing and testing programs.

Example: You can use the CMS READCARD and MOVEFILE commands to create CMS files from source programs or existing JCL statements that are on cards or magnetic tape. One advantage of storing source programs on CMS disks is that they can be maintained as backup copies of a program while a second version is being tested and debugged. By using the System Product Editor or commands like COPYFILE, SORT, and RENAME, you can modify and copy CMS disk files.

Refer to the *VM/SP CMS User's Guide* for information on how to compile and run many types of OS/VS programs under CMS.

OS Uniprocessor Under VM/SP

When operating MVS in uniprocessor mode under VM/SP, VM/SP simulates three privileged and two nonprivileged System/370 instructions.

The three privileged instructions are:

CLRIO	(clear I/O)
IPK	(insert PSW key)
SPKA	(set PSW key from address)

The two nonprivileged instructions are:

CS	(compare and swap)
CDS	(compare double and swap)

VM/SP allows the compare instructions (CS and CDS) to execute normally; it does not simulate them when the real machine is equipped with the appropriate hardware feature. However, when MVS is run under VM/SP on a machine that does not have these instructions installed, VM/SP simulates them.

Use of Single Processor Mode in AP and MP Systems

In tightly-coupled multiprocessing (MP) and attached processor (AP) systems, single processor mode allows you to dedicate a processor to an MVS V=R virtual machine. In single processor mode, VM/SP runs in uniprocessor mode in the main processor, and the MVS V=R virtual machine runs under VM/SP in the main processor and has the exclusive use of the other processor for MP or AP operations. An MVS V=V virtual machine cannot use single processor mode. However, other virtual machines can operate under VM/SP concurrently with the MVS V=R virtual machine in single processor mode. Prior to single processor mode, MVS virtual machines could only run MVS in uniprocessor mode -- not in MP or AP mode.

Operating Procedures

To use single processor mode, you must meet these four conditions:

1. The multiprocessing feature is installed on the hardware for the MP or AP system.
2. VM/SP has a V=R storage area.
3. VM/SP is running in uniprocessor mode.
4. The system operator enables single processor mode for the VM/SP system by issuing the class A CP command:

```
spmode on
```

Note:

- To run in uniprocessor mode if VM/SP was generated to support an AP or MP system, the system resource operator (class B) must vary offline the attached processor. Once it is offline, VM/SP begins running in uniprocessor mode.
- If you are using single processor mode on a 3081 processor, use the VARY OFFLINE PROCESSOR VLOG command to logically vary the processor offline. If the FORCE or the VPHY option of the VARY command is used, the processor will be physically varied offline to the configuration and is unavailable to the MVS virtual machine.

After VM/SP is in single processor mode, the MVS virtual machine user must IPL (or re-IPL) MVS so that it can gain control of the dedicated processor for MP or AP operations.

Note: When you initialize the MVS V=R virtual machine before VM/SP is in single processor mode, you are initializing MVS in uniprocessor mode -- not in MP or AP mode.

In single processor mode, VM/SP simulates MP privileged instructions issued by the MVS virtual machine. It also reflects MP external interruptions to this virtual machine. When other interruptions (such as I/O) or privileged instructions occur in the main processor, VM/SP simulates them and reflects the activity to the proper virtual machine. When they occur in the dedicated processor, the MVS virtual machine handles them.

To leave single processor mode, the MVS V=R virtual machine user should first finish all MVS processing and reset the virtual machine. (The user can reset the virtual machine in several ways: by initializing CMS, by issuing the CP SYSTEM RESET command, or by logging off the virtual machine.) Once processing is finished and the virtual machine is reset, the VM/SP system operator issues the class A CP command:

```
spmode off
```

This command returns control to VM/SP in the normal uniprocessor mode of operation. If VM/SP was generated to support an AP system, VM/SP can resume AP operations in the attached processor when the system resource operator (class B) varies it online.

To determine whether the system is in single processor mode, either the system operator (class A) or a virtual machine user (class G) can issue the CP command:

```
query spmode
```

Restrictions

1. When the MVS virtual machine runs in MP mode, the MVS virtual machine operator should not vary offline the main processor. Varying MVS offline in the main processor would cause these problems:
 - a. VM/SP cannot control the main processor.
 - b. VM/SP abnormally terminates when the MVS user and the dedicated processor attempts to vary the main processor online.
 - c. MVS in native mode (through a dynamic SCP transition to native mode as previously described in this section) cannot return and operate as a virtual machine under VM/SP.
2. Single processor mode does not support either real or virtual channel reconfiguration. Thus, if an MVS system is to operate in a virtual machine and use single processor mode, do not generate the system with channel reconfiguration hardware (CRH) support. That is, do not specify `OPTIONS=(CRH)` in the MVS `CTRLPROG` system generation macro.
3. You can generate MVS to run in a virtual machine with TCAM active. However, you cannot use TCAM in the single processor mode environment if TCAM was generated with the `VM/370` option. Specifying the `VM/370` option with TCAM forces MVS to issue `DIAGNOSE` instructions when running in a virtual machine environment. This can cause undesirable results when MVS tries to execute the `DIAGNOSE` instructions on the native processor. Therefore, TCAM should not be generated with the `VM/370` option if the virtual MVS system is running in single processor mode. The `SET NOTRANS ON` command should be issued when running `V=R` and using TCAM.
4. Single processor mode does not support the MVS `QUIESCE` command. CP cannot control to which processor the MVS `V=R` virtual machine will dispatch the task to handle the MVS `QUIESCE` command. If the task gets dispatched on the MVS native processor, it will issue a stop sign and store status to the CP processor, thus putting it into a manual state. Also the registers from the stop and store status may not be those of the MVS `V=R` virtual machine; therefore, if the MVS `V=R` virtual machine tried to use them, the results would be unpredictable.
5. On a 3081 processor, the MVS `VARY PROCESSOR OFFLINE` command disconnects the channel set of the processor that was varied offline. That is, if an operator issues `VARY PROCESSOR OFFLINE` on a 3081 and the channel set of the real processor is identical to the channel set in the MVS Sysgen, the processor that was varied offline will lose its channels. When the operator sets `SPMODE` off and varies the processor back online, that processor will not have I/O capabilities. To reconnect the channels, the operator should use the reconfiguration frame on the 3081 before varying the processor online.
6. MVS must not vary VM/SP's storage online. This would cause unpredictable and disastrous results for VM/SP. That is, VM/SP would still be operating until MVS altered some of VM's storage.

7. MVS must not attempt to STOP or RESET VM/SP's processor. This would be an event that was not initiated by CP and there would be no possible way to recover, since it (unpredictably) stopped or reset VM's real processor.

Error Recording

When in single processor mode, VM/SP cannot intercept SVC 76 (the error recording SVC) in the dedicated processor. Thus, error records for the MVS V=R virtual machine are in two locations: (1) the V/M error recording cylinders when SVC 76 is issued in the main processor, and (2) the MVS SYS1.LOGREC data set when SVC 76 is issued in the dedicated processor.

To find all the error records that pertain to the MVS V=R virtual machine, you must look in both locations. To put the records in chronological sequence, you can follow the time and date recorded on each record.

Note: Duplicate error records appear for channel checks reflected on the main processor. When MVS in the main processor issues SVC 76 for a channel check, VM/SP intercepts the SVC 76 and records the error in its error recording cylinders. However, VM/SP then reflects (or passes) the SVC 76 back to MVS for recording in its SYS1.LOGREC data set.

Taking a VM/SP Dump

In single processor mode, when the PSW restart key is pressed, VM/SP reflects the restart interruption back to the MVS V=R virtual machine and does not take a VM/SP dump.

To take a VM/SP dump while VM/SP is in single processor mode, the system operator must follow these steps:

1. Display the restart PSW at main storage location zero.
2. Subtract eight bytes from the instruction address in this PSW. (The instruction address is in the last three bytes of the PSW.)
3. Locate the storage area pointed to by the address determined in step 2 and store X'FF' in the first byte of that location. Do not change the other three bytes.
4. Press the restart key to take the dump.

After the dump is taken, the dump program automatically reinitializes VM/SP.

Note: When CP is *not* in a loop or wait state, the VM/SP system operator can use the CP commands DCP and STCP (class C) to perform these steps. Otherwise, the operator must perform these steps at the console. For details about how to use the console to display and alter main storage, refer to the appropriate System/370 operating procedures publication.

Summary

When loading OS/VS into a virtual machine, the terminal becomes the OS/VS operator console, and the virtual machine user becomes the operator responsible for entering all commands and responses. The four basic techniques for running OS/VS in a virtual machine are:

1. Batch mode
2. Alternating between OS/VS and CMS
3. In a single virtual machine
4. For OS/VS2 users only, running OS/VS2 disconnected.

Before using one of these techniques, you must understand how to:

- Generate OS/VS to run in a virtual machine
- Create VM/SP directory entries for OS/VS virtual machines
- Access the OS/VS system residence volume
- Ensure that the proper I/O devices are attached to the OS/VS virtual machine
- IPL and operate OS/VS under VM/SP

The primary objectives when generating OS/VS to run in a virtual machine should be to have all commonly used transient routines resident in storage and to run all jobs V=R if possible. To meet these objectives, you need to consider how you generate both VM/SP and OS/VS. (OS/VS can also be generated under VM/SP.)

To control OS/VS in a virtual machine, use OS/VS operator commands to hold and release queues and jobs, and to start initiators or define partitions. You can observe the progress of the command's execution by following the OS/VS messages. Also, additional operator commands and control statements must be entered at the console before running jobs on the OS/VS virtual machine.

OS/VS virtual machine users can use the System Product Editor and the EXEC facility to prepare jobs for execution in an OS/VS virtual machine. They can also use CMS commands to develop and test programs on CMS disks.

Index

Special Characters

#CP function, using 4-16

A

abend dump
 printing for virtual machines 1-5
abnormal terminations 1-5
accessing
 devices, by VM/SP virtual machine 2-5
 DOS 3-6
 error records, for operating systems under VM/SP 1-3
 virtual machines via AUTOLOG 1-56
ACCOUNT control statement 1-52
accounting
 elapsed processor time, in DOS/VS virtual machine 3-4
ACCT option 1-6
action routine, sample response file 1-16
address, virtual device 4-11
advanced functions-DOS program product 1-16
allocating space
 CP disks, for VM/SP virtual machines 2-3
 CP format/allocate program 1-24
alternate machine consoles, specifying for virtual machine 1-21
alternate path support
 for virtual machines 1-25
 mutually exclusive with reserve/release support 1-24
 restrictions 1-26
 summary of reserve/release support 1-27
 using RDEVICE and RCTLUNIT macro instructions 1-25
alternating techniques
 between virtual machine operating systems 1-30
 between CMS and DOS under VM/SP 3-21
 between CMS and OS/VS under VM/SP 4-17
analysis of VM/SP problems 1-4
application program
 developing and testing 1-30
 alternating between operating systems 1-30
 programs to run in an OS/VS virtual machine 4-24
 timer-driven, specifying REALTIMER option for 1-8
 VIRT=REAL option, using for 1-9
attach processor and multiprocessor environment 1-20
attached processor systems 4-25
AUTOLOG facility 1-56
 AUTOLOG1 virtual machine and PROFILE EXEC 1-57
 defining AUTOLOG1 in the directory 1-56
 multiple systems with AUTOLOG1 1-57
 using the CP AUTOLOG command 1-56
autopoll channel programs, BTAM
 bypassing when using 1-12
 CCW change detection 4-8

B

batch
 mode 3-20
 using DOS in 3-20
 using OS/VS in 4-16
 virtual machine operating systems supported under VM/SP 1-3
BMX option 1-11
BTAM
 autopoll channel programs 1-12

C

channel
 BMX option, using 1-11
 BTAM autopoll 1-12
 dedicated, specifying 1-23
 exceptions 1-20
 model dependencies 1-19-1-20
 error recovery procedures 1-19
 overlapping SIO requests on 1-11
 programs 1-7
 ISAM 1-7
 virtual/real device relationships 1-19
 virtual/real relationships 1-19
CMS editor, using 3-22
CMS EXEC procedure
 OS/VS operation, example 4-15
CMS/DOS, linkage editor, program execution under CMS/DOS 3-4
common area, calculating for OS/VS2 virtual machine 1-42-1-44
communicating between virtual machines
 using IUCV 1-11
 using VMCF 1-11
communication lines, defining, using SPECIAL control statement 1-55
communications system, testing
 using multiple-access virtual machines 1-37
configuration, virtual machine, verifying 2-7
configurations for alternating between operating systems 1-32
console
 changing address 4-14
 redefining 2-8
CONSOLE control statement 1-52
CONSOLE control statement
 defining the operator's console 4-12
control program (CP), VM/SP component 1-1
conversational monitor system 1-1
 accessing 3-27
 alternating between OS/VS under VM/SP 4-17
 CMS file, using READCARD command to create 4-19
 CMS PUNCH command, using 4-18
 disconnecting 3-28
 function 1-3
 line mode required 1-29
 loading into a virtual machine 4-17
 OS/VS job streams, preparing 4-18
 returning to 3-29
 VM/SP component 1-1
CP commands
 #CP commands 3-19

- AUTOLOG command, using 1-56
- CLOSE 4-13
- entering 3-17
 - from DOS/VS virtual machine 3-17
 - from OS/VS virtual machine 4-16
- INDICATE 1-49
- LINK 4-11
- LOCATE 1-44
- MONITOR 1-49
- QUERY 3-7
- SET ECMODE 4-13
- SPOOL 3-31
- CP disks, formatting and allocating space for VM/SP virtual machine 2-4
- CP function, for OS/VS virtual machine, how to use 4-16
- CPTRAP, CP command 1-4
- CPU time, real 1-8

D

- data transfer
 - using IUCV 1-11
 - using VMCF 1-11
- DEDICATE control statement 1-53
- dedicated devices 1-4
- device/minidisk sharing 1-26
- DIAGNOSE instruction codes 1-11
- diagnose interface 1-17
- directory definition, VM/SP 2-1
- directory entry
 - creating VM/SP 1-50
 - multiple-access virtual machine 1-58
 - sample VM/SP 1-58
 - specifying virtual devices in 4-11
- directory entry considerations
 - control statements 1-52
 - ACCOUNT 1-52
 - CONSOLE 1-52
 - DEDICATE 1-53
 - IPL 1-52
 - LINK 1-54
 - MDISK 1-52
 - OPTION 1-52
 - SPECIAL 1-55
 - SPOOL 1-53
 - USER 1-52
- directory entry examples
 - alternating between operating systems 1-32
 - AUTOLOG1 virtual machine 1-57
 - MFT virtual machine 4-9
 - multiple-access TSO system 1-58
 - multiple-access virtual machine 1-35
 - running VM/SP under VM/SP 1-35
 - MVS virtual machine 4-10
 - PROFILE EXEC 1-57
 - secondary users controlling VSE/AF 3-5
 - TESTSYS 2-1
 - VSE/AF guest machine 3-5
 - VS1 virtual machine 4-10
- disconnection considerations
 - console READs and WRITEs 3-31
 - disconnected DOS machine
 - sending jobs to a 3-30
- display operator console 1-21
- DOS 1-16
 - accessing 3-6
 - accounting, elapsed processor time under VM/SP 3-4
 - alternating between CMS and DOS 3-21
 - basic running techniques 3-1
 - batch, running under VM/SP 3-20
 - card reader empty 3-24
 - CP, communicating with 3-19
 - directory entries, sample 3-5
 - DOS accounting 3-4
 - DOS recommendations 3-2
 - facilities available in CMS/DOS 3-32
 - generating a DOS supervisor 3-2
 - specifying boundary size 3-2
 - generating POWER 3-3
 - generating the operating system 3-3
 - handshaking 1-16, 3-2
 - in a virtual machine 3-1
 - initializing disks and minidisks 3-2
 - initializing DOS 3-23
 - interrupting DOS execution 3-20
 - loading DOS 3-8
 - automatic system initialization 3-8
 - IPL from card reader 3-17
 - IPL from console 3-13
 - logging onto 3-28
 - logical end line character 3-1
 - operation 3-18
 - operator's console, defining 3-7
 - output, examining 3-25
 - preparing job for 3-7
 - primary objectives 3-1
 - programs under DOS and CMS/DOS, executing 3-4
 - programs, developing and testing 3-31
 - punching CMS files 3-23
 - reading next spool reader file 3-24
 - reducing privileged instructions 3-2
 - reloading CMS 3-25
 - saved system 3-13
 - SLD directory reading, reducing 3-4
 - specifying DOS partition sizes 3-3
 - starting a job stream 3-18
 - summary 3-33
 - system generation recommendations 3-1
 - system residence volume, sharing 3-4
 - techniques for running 3-1
 - userid 3-1
 - primary 3-1
 - secondary 3-1
 - using CMS editor 3-22
 - using EXEC procedures 3-25
 - virtual unit record devices 3-6
 - card punch 3-6
 - card reader 3-6
 - controlling 3-7
 - issuing spool commands to control 3-23
 - printer 3-6
 - virtual device address 3-6
 - VM/SP recommendations 3-2
 - VM/SP saved systems 3-2
 - when job is finished 3-19
- DOS/VS
 - advanced functions-DOS program product 1-16
 - BTAM autopoll channel programs 1-12
- dump
 - printing virtual machine abends 1-5
 - processing 4-28
- dynamic processor reconfiguration 1-20
 - attach processor and multiprocessor environment 1-20
 - monitoring and service support facility 1-20
 - MSSFCALL diagnose instruction 1-20
 - Using the SCPINFO command 1-20
 - VARY PROCESSOR commands 1-20
 - 3081 processor complex 1-20

3082 processor controller 1-20
dynamic SCP transition to or from native mode 4-19
restriction for MVS virtual machine 4-20

E

ECMODE option 1-6
error records, writing 1-3

F

favored execution option 1-48
field developed programs
 used under VM/SP (reference) 1-3
first level storage 1-5
full screen console support 1-28

G

generation procedures under VM/SP 1-49
guest operating system
 running a 1-59

H

handling VS1 operator responses
 sample action routine 1-14
handshaking
 activating VM/VS for VS1 4-5
 DOS 1-16, 3-2
 VM/VS 1-7, 1-16
 VS1 1-17, 4-2, 4-4
high-water mark 1-39

I

I/O management, virtual machine 1-23
 CP format/allocate program 1-24
 CP format/allocate utility (DMKFMT) 1-24
 dedicated channels 1-23
 dedicated devices as an alternative to 1-24
 effects of 1-23
 need for 1-23
 defining direct access storage devices 1-24
 using as dedicated volumes 1-24
 using as minidisks 1-24
 device support facility program 1-24
 logical data sharing 1-24
 physical pack sharing 1-24
 T-disk space 1-24
 temporary disk space 1-24
 using the CP LINK command 1-24
 using the LINK directory control statement 1-24
 virtual disk device sharing 1-23
I/O waits 1-17
installed user programs
 used under VM/SP (reference) 1-3
inter-user communication vehicle
interactive response times 1-49
interval timer 1-8
IPL

a virtual machine, example 3-10
CMS 3-25
from the card reader 3-17
from the console 3-13
using a saved system 3-13
using ASI 3-8, 3-10

IPL control statement 1-52
IPL of VS1, sample 4-15
ISAM option 1-7

J

job streams (OS/VS), preparing
 using the CMS editor 4-18

L

LINK control statement 1-54
locked pages option 1-48
logical data sharing 1-24
Logon procedure, example 2-7

M

magnetic tapes 1-53
MDISK control statement 1-52
monitoring and service support facility (MSSF) 1-20
multiple consoles, specifying for virtual machine 1-21
multiple guest operating systems
 controlling 1-12
multiple shadow table support 1-38
multiple-access
 programs 1-33
 systems 1-2
 command language used 1-3
 dial-up terminals supported 1-36
 gaining access to VM/SP 1-3
 interactive users 1-3
 using under VM/SP 1-2
 virtual machine 1-58
 virtual machines 1-33
 a communications test system 1-37
 performance considerations 1-38
 virtual lines 1-33
 virtual transmission control unit 1-37
 virtual VM/SP illustrated 1-35
multiprocessing systems 4-25
multiprogramming systems 1-3
 under VM/SP 1-16
 diagnose interface 1-17
 special considerations 1-16
 VM/VS handshaking 1-16

N

native mode 4-19
 error recording 4-22
 transferring control of operating system to 4-20

O

- one-megabyte shadow tables, elimination of 1-39
- operating systems
 - alternating between 1-30
 - batch 1-2
 - configurations for alternating between 1-32
 - conversational 1-2
 - establishing priority 1-17
 - multiple access 1-2
 - multiple guest, controlling 1-12
 - native mode 4-20
 - page waits 1-17
 - performance guidelines 1-45
 - single-user interactive 1-2
 - transfer control of
 - from VM/SP to native mode 4-20
 - transferring output 1-30
 - VM/VS handshaking 1-7
- operator's console 4-12
- OPTION control statement 1-52
- OS/VS
 - #CP function, using the 4-16
 - accessing 4-10
 - alternating between CMS under VM/SP 4-17
 - batch mode under VM/SP 4-16
 - CMS EXEC procedure, example 4-15
 - CMS files, punching 4-18
 - communicating with CP 4-16
 - console READs and WriteS in a disconnected machine 4-23
 - with single console image facility 4-24
 - without single console image facility 4-23
 - CP commands, issuing 4-16
 - disconnected OS/VS machine 4-23
 - console READS and WRITEs in a 4-23
 - sending jobs to 4-23
 - disconnection considerations 4-23
 - examining virtual machine output 4-19
 - extended control mode 4-13
 - in a virtual machine 4-1
 - alternating technique 4-1
 - batch mode 4-1
 - console attention interrupt 4-2
 - CP spool files, closing 4-7
 - disconnected 4-1
 - disconnected user (OS/VS2 only) 4-1
 - generating VM/SP 4-2
 - handshaking for VS1 4-2
 - IPL command enhancement 4-2
 - miscellaneous enhancements 4-8
 - saved systems 4-2
 - sharing system residence volume 4-3
 - specifying options 4-3
 - system generation 4-1
 - VS1 without handshaking 4-8
 - initializing 4-19
 - job entry 4-13
 - job streams, using CMS editor to prepare 4-18
 - loading CMS 4-17
 - loading OS/VS 4-13
 - multiple userids, using 4-13
 - MVS console screen image, saving 4-14
 - operation 4-14
 - operator's console, defining 4-12
 - output retrieval 4-13

- preparing jobs for virtual machine 4-12
- printing abend dumps under VM/SP 1-5
- programs run in an OS/VS virtual machine
 - developing and testing 4-24
- punching CMS files 4-18
- reloading CMS 4-19
- sample directory entries 4-9
 - a VS1 virtual machine 4-10
 - an MFT virtual machine 4-9
 - an MVS virtual machine 4-10
- spool commands, issuing 4-18
- summary 4-29
- system residence volume
 - attaching to userid 4-11
 - defining as read/write disk 4-11
 - defining location of 4-10
 - using CP LINK command 4-11
- unit record devices, controlling 4-18
- virtual devices, using 4-11
- virtual machine storage size 4-14
 - altering 4-14
 - definition limits 4-14
- VM/SP spool file system, using 4-12

P

- page fault 1-5
- paging activity
 - reducing 1-5
- paging environment 1-4
- performance
 - configuration factors influencing 1-45
 - guidelines 1-45
 - I/O waits, establishing processing priority 1-17
 - measurements 1-49
 - page waits 1-17
 - reducing paging 1-5
 - virtual machine, paging factors affecting 1-5
 - VM/SP factors 1-47
 - workload factors influencing 1-46
- performance/monitor analysis program (VMAP) 1-43, 1-47
- physical pack sharing 1-24
- printing dumps
 - OS/VS abend dump under VM/SP 1-5
- priority 1-48
 - establishing for operating systems 1-17
 - I/O waits 1-17
 - page waits 1-17
- processor model dependencies 1-19-1-20
 - channel error recovery procedures 1-19
- program products
 - factors affecting performance 1-5
 - used under VM/SP (reference) 1-3
- programmable operator facility
 - action routine 1-16
 - automatic execution 1-12
 - handling VS1 operator responses 1-14
 - in a standalone VM/SP system 1-12
 - operation described 1-12
 - sample action routine 1-14
 - sample response file 1-16
 - systems designed for 1-12
 - use in a single system 1-13
 - use in distributed systems 1-14

R

- READCARD command 4-19
- real CPU time 1-8
- real device equivalents, requirement for 1-4
- REALTIMER option 1-8
- reserve/release support
 - definition of 1-24
 - for dedicated volumes 1-27
 - mutually exclusive with VM/SP alternate path support 1-24
 - operating systems using DASD 1-25
 - special conditions 1-25
 - types of support 1-25
 - shared DASD 1-25
 - virtual reserve/release 1-25
 - using CCW commands 1-25
- reserved page frames 1-48
- resources, virtual machine 1-1
 - allocating 1-1
- RSCS networking program 1-2

S

- sample OS/VS directory entries 4-9
- second level storage 1-5
- secondary console, specifying 1-21
- secondary userid, specifying 1-22
- segment table origin control block 1-38
- selective invalidation 1-39
- SET BYPASS nnnnk command 1-39
- SET RUN ON CP command 1-12
- SET STBYPASS command 1-40
- SET STMULTI command 1-39, 1-42
- shadow table bypass for V=R users 1-39
- shadow table bypass for V=V users 1-40
- shadow table maintenance support
 - multiple shadow table support 1-38
 - segment table origin control block 1-38
 - one-megabyte shadow tables, elimination of 1-39
 - selective invalidation 1-39
 - shadow table bypass for V=R users 1-39
 - SET STMULTI command 1-39
 - shadow table bypass for V=V users 1-40
 - restrictions using shadow table bypass 1-42
 - SET STBYPASS command 1-40
 - SET STMULTI command 1-42
 - STFIRST directory option 1-42
- shadow tables
 - restrictions when eliminating 1-39
- shared DASD
 - CCW commands 1-25
 - data protection 1-25
 - special considerations 1-25
- single processor mode 4-25
 - condition requirements 4-25
 - error recording 4-28
 - leaving single processor mode 4-26
 - operating procedures 4-25
 - restrictions 4-27
 - VM/SP dump 4-28
- single-user interactive system 1-3
- SIO instructions
 - options
 - specifying 1-6
 - reducing number executed 1-6
- SPECIAL control statement 1-55
- spool commands, using 4-18
- SPOOL control statement 1-53

- spool file system, using 4-12
- spooling
 - closing files 1-19
 - recommendations 1-18
 - what spooling to use 1-18
- STFIRST directory option 1-42
- STFIRST option 1-8
- summary of reserve/release support 1-27
- SVCOFF option 1-9
- system control program (SCP) 4-19
- system product editor 3-22
- system residence volume 3-6, 4-10
 - ways made available 3-6
 - attach to userid 3-6
 - define as read/write disk 3-6
 - link to DOS system residence volume 3-6

T

- T-disks, defining and using for virtual machines 1-24
- tape devices, summary of VM/SP reserve/release support 1-28
- TEST BLOCK instruction 1-10
- third level storage 1-5
- time of day 1-8
- time-of-day clock 2-5
- time, real CPU 1-8
- timer, interval 1-8
- trace table recording facility
 - analysis of VM/SP problems 1-4
- transferring output 1-30
- TRAPRED, CMS utility program 1-4

U

- unit record devices 1-53
- unit record output, reducing 4-17
- unsupported devices 1-4, 1-54
- USER control statement 1-52

V

- virtual devices
 - address, changing 4-11
 - defining 1-55
 - direct access storage devices (OS/VS) 4-11
 - remote terminals 1-34
 - specifying in VM/SP directory entry 4-11
 - states or conditions 1-23
 - dedicated 1-23
 - shared 1-23
 - spooled 1-23
 - using with OS/VS in virtual machine 4-11
- virtual card reader (OS/VS) 4-11
- virtual printer (OS/VS) 4-11
- virtual punch (OS/VS) 4-11
- virtual tape (OS/VS) 4-11
- virtual IPL and operation 2-5
 - accessing devices 2-5
 - graphic and spool devices 2-5, 2-6
- virtual disks 2-5, 2-6
- virtual machine
 - #CP function, using 4-16
 - abend dump 1-5
 - abnormal terminations 1-5

- accessing a VM/SP system 2-3
- batch 1-3
- BMX option 1-11
- CMS system 2-3
- communicating between
 - using IUCV 1-11
 - using VMCF 1-11
- configuration 2-2
- CP disks 2-3
 - allocating spaces 2-4
 - formatting the volumes 2-3
- data transfer between 1-11
 - using IUCV 1-11
 - using VMCF 1-11
- defining a console for VM/SP 2-3
- definition of 1-1
- DOS in a virtual machine 3-1
- dump, printing abends 1-5
- extended control-program support
 - effect on paging performance 1-5
- guest operating system, running a 1-59
- handshaking 1-7
 - operating systems supported 1-7
- I/O instructions, reducing number 1-5
- loading CMS 3-22
- multiple access 1-2
- MVS restrictions 4-27
- operating systems 1-2
- options
 - ACCT 1-6
 - BMX 1-11
 - ECMODE 1-6
 - ISAM 1-7
 - REALTIMER 1-8
 - STFIRST 1-8
 - SVCOFF 1-9
 - Virtual=Real 1-9
 - VMSAVE 1-10
 - 370E 1-11
- paging
 - reducing 1-5
- printing dumps for operating systems 1-5
- reducing paging 1-5
- reducing VM/SP paging activity for 1-5
- resources, allocating 1-1
- SIO instructions
 - reducing number executed 1-6
- SIO instructions, reducing I/O operations for 1-5
- spooling considerations 2-6
- using more than one 3-27, 4-22
- VM/SP in a virtual machine 2-1
 - CMS system residence volume 2-3
- VM/SP in VM system, summary of 2-20
- VS1, generating 4-3
 - activating VM/VS handshaking 4-5
 - nonpaging mode 4-5
 - storage limits 4-3
 - VM/VS handshaking for VS1 4-4
 - VS1 in V=R 4-4
- 3850 mass storage system considerations 4-9
- virtual machine assist
 - effect on virtual machine paging performance 1-5
- virtual machine communication facility
 - data transfer 1-11
 - DIAGNOSE instruction codes 1-11
 - notifying VM/SP of BTAM autopoll channel programs 1-12
- virtual machine configuration, verifying 2-7
- virtual machine consoles 1-21
 - consoles supported by operating systems 1-21
 - alternate consoles 1-21
 - multiple consoles 1-21
 - specifying 1-21
 - specifying a secondary userid 1-22
 - specifying local 3270 as primary console 1-22
 - specifying the second console 1-21
 - using a display terminal as a console in display mode 1-22
 - using specific devices as virtual consoles 1-21
 - using the CP DEFINE GRAF command 1-22
 - using the SPECIAL statement 1-22
- virtual machine environment
- virtual reserve/release support
 - benefits to other systems 1-26
 - effects on same minidisk users 1-26
 - example 1-27
 - how to specify 1-26
 - restrictions 1-26
 - defining alternate paths 1-26
 - device sharing between real processors 1-26
 - device/minidisk sharing on a single processor 1-26
 - preservind data integrity 1-26
 - sharing minidisks 1-26
 - using ALTCH macro 1-26
 - using ALTCU macro 1-26
 - using more than one minidisk 1-26
 - summary 1-27
- virtual unit record devices 3-6
- Virtual=REAL option 1-9
- VM/IPCS extension program 1-2
- VM/SP (Virtual Machine/System Product)
 - BTAM autopoll programs 1-12
 - channel programs, ISAM 1-7
 - creating directory entries 1-50
 - debugging, printing dumps 1-5
 - directory definition 2-1
 - ECMODE option
 - See ECMODE option
 - expanding capabilities of
 - RSCS networking program 1-2
 - VM/IPCS extension program 1-2
 - general considerations 1
 - generation procedures 1-49
 - handshaking 1-47
 - I/O instructions, reducing number 1-5
 - ISAM option
 - See ISAM option
 - multiple access virtual machine 1-2
 - multiprogramming systems 1-16
 - page waits 1-17
 - paging environment 1-4
 - paging factors
 - effect of application programs 1-5
 - performance measurements 1-49
 - performance options 1-47
 - printing abend dumps 1-5
 - printing OS/VS abend dumps 1-5
 - priority for operating systems, establishing 1-17
 - reducing paging 1-5
 - shadown table maintenance support 1-38
 - storage levels
 - first 1-5
 - printing dumps for 1-5
 - second 1-5
 - third 1-5
 - uniprocessor, OS 4-25
 - virtual devices, processing I/O operations 1-23
- VM/SP components
 - control program (CP) 1-1
 - conversational monitor system (CMS) 1-1

VM/SP under VM/SP, example 2-7-2-20
VM/VS handshaking 1-7
 VSE-VM/370 linkage facility 1-16
VMSAVE option 1-10
VSE-VM/370 linkage facility 1-16
VS1 nonpaging mode 4-5
 FASTNIP forcing of 4-6
 minimum size VS1 nucleus, providing a 4-6
 storage limits 4-6
 V=R jobs, executing 4-6
 virtual storage size message, response to 4-6
 VS1 paging data set 4-6
 VS1 storage mapping 4-6
VS1 operator responses 1-14
 sample action routine 1-14
VS1, handshaking for 1-17

W

workload factors influencing performance 1-46

2

2305 devices, dedicating to virtual machine 1-54

3

3081 processor complex
 monitoring and service support facility (MSSF) 1-20
 MSSFCALL diagnose instruction 1-20
 processor reconfiguration 1-20
 SCPINFO command 1-20
 shadow table usage 1-39
 single processor mode 4-26
 TEXT BLOCK instruction 1-9
 validating storage 1-9
3082 processor controller
 monitoring and service support facility (MSSF) 1-20
 processor reconfiguration 1-20
370E option 1-11
370E, using with virtual machine assist 1-11



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.*

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

	Yes	No
• Does the publication meet your needs?	<input type="checkbox"/>	<input type="checkbox"/>
• Did you find the material:		
Easy to read and understand?	<input type="checkbox"/>	<input type="checkbox"/>
Organized for convenient use?	<input type="checkbox"/>	<input type="checkbox"/>
Complete?	<input type="checkbox"/>	<input type="checkbox"/>
Well illustrated?	<input type="checkbox"/>	<input type="checkbox"/>
Written for your technical level?	<input type="checkbox"/>	<input type="checkbox"/>
• What is your occupation? _____		
• How do you use this publication:		
As an introduction to the subject?	<input type="checkbox"/>	As an instructor in class? <input type="checkbox"/>
For advanced knowledge of the subject?	<input type="checkbox"/>	As a student in class? <input type="checkbox"/>
To learn about operating procedures?	<input type="checkbox"/>	As a reference manual? <input type="checkbox"/>

Your comments:

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

VM/SP Operating Systems in a Virtual Machine (File No. S370/4300-34) Printed in U.S.A. GC19-6212-2

Fold and Tape

Please Do Not Staple

Fold and Tape

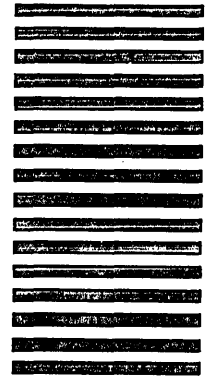


NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

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 _____



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.

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

	Yes	No
• Does the publication meet your needs?	<input type="checkbox"/>	<input type="checkbox"/>
• Did you find the material:		
Easy to read and understand?	<input type="checkbox"/>	<input type="checkbox"/>
Organized for convenient use?	<input type="checkbox"/>	<input type="checkbox"/>
Complete?	<input type="checkbox"/>	<input type="checkbox"/>
Well illustrated?	<input type="checkbox"/>	<input type="checkbox"/>
Written for your technical level?	<input type="checkbox"/>	<input type="checkbox"/>
• What is your occupation?	_____	
• How do you use this publication:		
As an introduction to the subject?	<input type="checkbox"/>	As an instructor in class? <input type="checkbox"/>
For advanced knowledge of the subject?	<input type="checkbox"/>	As a student in class? <input type="checkbox"/>
To learn about operating procedures?	<input type="checkbox"/>	As a reference manual? <input type="checkbox"/>

Your comments:

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

VM/SP Operating Systems in a Virtual Machine (File No. S370/4300-34) Printed in U.S.A. GC19-6212-2

Fold and Tape

Please Do Not Staple

Fold and Tape

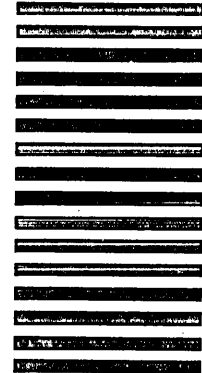


NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

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 _____

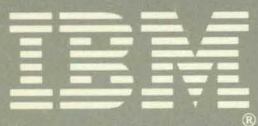
State _____ Zip Code _____

IBM Branch Office serving you _____



GC19-6212-2

VM/SP Operating Systems in a Virtual Machine (File No. S370/4300-34) Printed in U.S.A. GC19-6212-2



GC19-6212-2

