VM For MVS Systems Programmers, Part 2

Martha McConaghy, Marist College
Mark Post, Novell

Monday, March 2, 2009
Session 9128

Follow Up Presentations

- Other presentations this week which cover related subjects in more detail:
  - 9102 The Very Basics of z/VM – Concepts and Terminology Mon 9:30am (available on proceedings)
  - 9115 VM Performance Introduction Mon 1:30pm (available on proceedings)
  - 9107 – 9108 Introduction to VM Hands-on Lab Tue 8am-10:30
  - 9117 Introduction to VMSES/E for z/VM Thur 8:00am
  - 9118 Maintaining z/VM with VMSES/E demo Thur 9:30am
  - 9124 Experiences using z/VM VSWITCH Tue 1:30pm
  - 9129 z/VM Security and Integrity Wed 3pm
  - 9133 Configuring, Customizing and Modifying your VM System without an IPL Tue 1:30pm
  - 9134 Dynamically Managing Hardware I/O Configuration using VM Tue 3pm

Agenda

- Part 1:
  - Introduction to and comparison of basic concepts
  - What is a hypervisor and what about all the “virtual” stuff?
  - Caring for a VM system (maintenance, system datasets, etc.)
  - System configuration concepts
- **Part 2:**
  - **Applications and guests**
  - **CMS vs. TSO**
  - **File Editors**
  - **Misc good stuff**

We will answer questions as time allows…..

Application Support

- CP level of z/VM cannot execute applications. It's only job is to emulate hardware and manage resources.
  - Does not provide file editing or writing capabilities. It can read CMS files for use with system configuration.
  - Cannot compile, assemble or execute programs.
- CMS started out as an operating system during the early days of mainframes.
- Became popular as a "guest" on CP to provide user/application functions. It became the standard for later versions of VM and can no longer run outside of the CP environment.
- Most VM based applications still run in CMS, though some are now showing up on Linux as well.

Application Support

- VM does not have true BATCH capabilities. CMS is a single user environment, like a PC. Programs are either executed interactively with the user, or while the virtual machine is running disconnected.
- Virtual machines being used as “servers” normally run in “disconnected” mode, i.e. without a physical console. They are “autologged” at the IPL of CP, or shortly thereafter.
  - Known as “service virtual machines” or SVMs.
- Similar to started tasks (STCs) on MVS.

Application Support

- When a CMS user executes a program, it runs in the virtual storage (or address space) of the virtual machine.
- Virtual Machines can communicate with each other through several methods. Each of these methods are provided by CP:
  - IUCV (Inter User Communication Vehicle) – vm to vm communication
  - Socket – basic TCP/IP communication with users or other services
  - Guest LAN/Vswitch – as if virtual machines were servers on a network
  - CTCA (channel to channel) – old fashioned, but still works
  - Spool files (as old-fashioned as it gets, but very useful!)

Saved Segments

- In some cases, an application may require a lot of virtual storage in order to run. This becomes a problem if many virtual machines need to run the same application.
- Discontiguous Saved Segments (DCSS) allow many virtual machines to share all or part of an application's code, thus saving on memory requirements. The concept is similar to the Link Pack Area in z/OS.
- For example, QMF (Query Management Facility – for DB2) runs on both z/OS and VM. When the VM version is installed, parts of the QMF code are loaded into a DCSS. This is then shared by all virtual machines that start up the QMF product. This means that each of the virtual machines can have less virtual storage and still be able to run QMF.

Named Saved Systems

- A Named Saved System is similar to a DCSS. Both reside in CP SPOOL. Both can be shared by many virtual machines.
- While a DCSS contains all or part of an application, a NSS contains all or part of an operating system.
- CMS is the most important user of NSS at this time.
- Rather than each CMS virtual machine loading its own, unique copy of CMS, a shared copy is loaded. This shared copy is stored in an NSS, usually called CMS.
- Usually only the most commonly used parts of a system are stored in a NSS.
- Linux also has the capability of using NSS and DCSS.

Saved Segments - Example

FILE	FILENAME	FILETYPE	MINSIZE	BEGPAG	ENDPAG	TYPE	CL	#USERS	PARMREGS	VMGROUP
0244	SCEEX	DCSS	N/A	02100	027FF	SR	A	00039	N/A	N/A
0250	CMSBAM	DCSS-M	N/A	00B0D	00B37	SR	S	00000	N/A	N/A
0249	DOSBAM	DCSS-S	N/A	00B00	00B37	--	S	00000	N/A	N/A
0245	NLSAMENG	DCSS	N/A	02A00	02AFF	SR	A	00157	N/A	N/A
0240	CMSFILES	DCSS	N/A	01900	01BFF	SR	A	00000	N/A	N/A
0239	SVM	DCSS	N/A	01900	019FF	SR	A	00001	N/A	N/A
0232	GCS	NSS	0000256K	00000	0000C	EW	R	00002	OMITTED	YES
				00400	0044E	SR				
				0044F	0044F	SW				
				00450	005FF	SN				
				01000	0101A	SR				
				0101B	011FF	SN				
0233	CMS	NSS	0000256K	00000	0000D	EW	A	00196	00-15	NO
				00020	00023	EW				
				00F00	013FF	SR				

Comparing
CMS
Versus
TSO

CMS vs. TSO

- CMS is command line based. Many utilities have full screen interfaces, but a lot of time is spent with the command line.
- ISPF exists, but is not widely used. It is required for products ported from z/OS such as QMF. Most CMS utilities with full screen interfaces, such as FILELIST, use other screen management tools.
- CMS reads and writes sequential files, rather than datasets. Files reside on either minidisks or in Shared File System (SFS) directories.
  - Shared File system provides pooled filespace which can be more easily managed than minidisks.
- A CMS file = sequential dataset in TSO.
- Nothing in CMS comparable to a PDS in TSO.

CMS vs. TSO

- Minidisks and SFS directories have to be “accessed” in order to be used. The access command assigns an alphabetical letter, which is used to determine search order for a variety of purposes.
- Search order starts with A mode and goes downward to Z. Similar to “PATH” in UNIX/Linux.
- **QUERY SEARCH** command will show you the order of your disks and directories.

```
q search
-       DIR  A    R/W   ACADEM:URMM.
MAROPR  39F  B    R/O
MARIST  19F  C    R/O
-       DIR  D    R/W   ACADEM:URMM.MAILTOOLS
TCM592  390  E    R/O
-       DIR  F    R/W   ACADEM:URMM.WEBSHARE.RESNET
DB2195  391  G    R/O
-       DIR  J    R/O   ACADEM:URMM.TOOLS
-       DIR  K    R/O   ACADEM:URMM.QIP2
GDDM32  319  P    R/O
-       DIRC R    R/O   ACADEM:UIFB.PUB
MNT190  190  S    R/O
VSP510  510  T    R/O
MNT19E  19E  Y/S  R/O
```

CMS vs. TSO

- File identifiers in CMS are in the format:

  Filename Filetype Filemode

  FILE1 INFO A

  - **Filename** can be 1 to 8 alphanumeric characters long.
  - **Filetype** can also be 1 to 8 alphanumeric characters. Some filetypes have special meaning, such as “EXEC”, “MODULE”, “TXTLIB”, etc.
  - **Filemode** is a single alphabetic character which indicates the access mode of the disk where the file resides. Therefore, this value can and will change if the disk is moved to a different access mode.
- Filemodes S and Y are special. They hold the CMS system disks and should not be changed.
- The mode of a file in an SFS directory can also contain the names of the root and subdirectories:

  FILE1 INFO VMSYS:MYACCT.SUB1

CMS vs. TSO

- CMS has no concept of a catalog or “high level qualifier”.
- Files are accessible as long as the minidisk or directory that holds them is accessible to the virtual machine and to CMS.
- Security is provided by controlling which virtual machines can link which minidisks or access which SFS directories.
  - Minidisk linking is controlled by CP via the Directory or an external security manager such as RACF or VM:Secure (CA).
  - SFS directory access is controlled by rules created by SFS administrator or owner of the directory.
- Because CMS runs in a virtual machine, it does not have any impact on the running CP level of VM.

CMS – Writing/Executing Programs

- Most common compilers available for CMS including LE
  - May not always be same levels as z/OS
- OS Binder/Loader available to build executables. LinkEdit used to build text libraries. (Again, not quite a PDS.)
- **REXX** is the most commonly used scripting language. Especially good for developing quick solutions to problems and challenges. REXX programs are called “execs” and are stored in filetype EXEC. **No CLISTS in CMS.**
- **CMS Pipelines** provides extremely powerful application development/data processing functions. Can be used within REXX execs or from command line. Much more “full function” than Pipelines on TSO.

CMS – Writing/Executing Programs

- Unlike TSO, CMS is not inherently multi-tasking.
  - Enhancements in recent years have made it possible to write multi-tasking programs in CMS environment.
  - IMAP server that comes with TCPIP is an example of a multi-tasking application running in CMS.
- Data for CMS applications can exist in various places depending on need:
  - CMS text file (on minidisk or SFS directory)
  - BFS file
  - DB2 tables
  - VSAM (yes, there is a VSAM, but it is VSE version)
  - SPOOL

Other Application Platforms

- VM Open Edition (OE) is similar to USS on MVS. It provides a POSIX compliant environment, except for “fork”.
  - Byte File system (BFS) is similar to HFS.
  - CMS can execute programs stored in BFS.
- Group Control System (GCS) is a separate component of z/VM. Provides multi-tasking shared application space for several products ported from MVS, i.e. VTAM and NETVIEW. Also required for RSCS.
- Linux is now being used to provide services for VM as well. For example, SSL service provided with TCP/IP on VM runs in a Linux virtual machine.

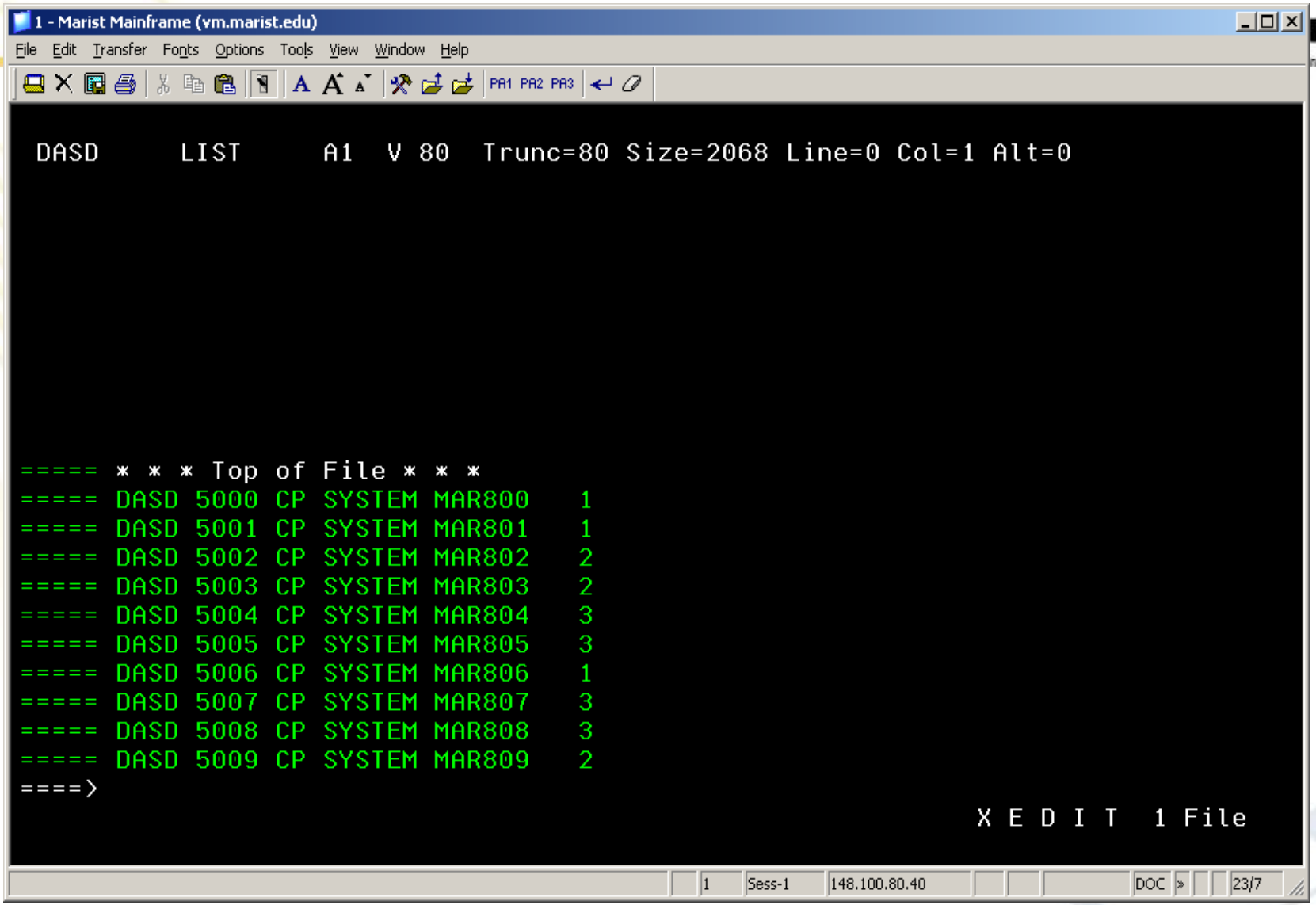
CMS Editor (XEDIT)

Vs.

TSO Editor

Editor Comparison

- **Xedit** is the CMS editor. It has a lot in common with TSO editor. Perhaps too much? Some of the commands are reversed.
  - **File = Save** and vice versa.
- Full screen editor like ISPF/PDF and provides prefix command area, but the line commands are different.
- More than 1 file can be edited at one time. They can be accessed in a “ring”, or can be displayed on screen at same time in split window mode.
- Highly customizable for individual taste. (Every “old time” VMer has a different Xedit profile.)
- Can be used to develop applications needing full screen interface. FILELIST and DIRLIST are two that come with CMS.


1 - Marist Mainframe (vm.marist.edu)
File Edit Transfer Fonts Options Tools View Window Help
PA1 PA2 PA3
DASD LIST A1 V 80 Trunc=80 Size=2068 Line=0 Col=1 Alt=0
===== * * * Top of File * * *
===== DASD 5000 CP SYSTEM MAR800 1
===== DASD 5001 CP SYSTEM MAR801 1
===== DASD 5002 CP SYSTEM MAR802 2
===== DASD 5003 CP SYSTEM MAR803 2
===== DASD 5004 CP SYSTEM MAR804 3
===== DASD 5005 CP SYSTEM MAR805 3
===== DASD 5006 CP SYSTEM MAR806 1
===== DASD 5007 CP SYSTEM MAR807 3
===== DASD 5008 CP SYSTEM MAR808 3
===== DASD 5009 CP SYSTEM MAR809 2
====>
X E D I T 1 File
1 Sess-1 148.100.80.40 DOC 23/7

Monitoring/Automating
CP
and
Virtual Machines

System Consoles

- Each virtual machine has a “system console” defined in the CP directory.
  - Most will be a 3215 console, which was a real device once upon a time. It was a line-mode device, which CP still emulates. CMS requires this type of console, but others find it useful too.
  - MVS and z/VM guests require 3270 type consoles.
- Virtual machines with line-mode consoles can have all messages written to a running log. This log is stored in spool, Similar to SYSLOG – but is unique to each virtual machine.
- The console spool file can be “sent” to another virtual machine, such as an archiver, or stored in spool for later use.
- Tools like TRACK can be used to peek at a live console without affecting the running application. Great for diagnosing problems!

Monitoring System and Applications

- The system operator receives all CP system messages. It can also receive messages from other virtual machines via SCIF (Secondary Console Interactive Facility) – a feature built into CP.
- SCIF allows the system operator to reroute incoming messages or take action based on them. An additional product is also required:
  - PROP (Programmable Operator) – comes with VM as part of CMS
  - VM:OPERATOR (CA product)
  - Operations Manager (IBM product)
- This function is similar to Netview on z/OS or syslogd on a UNIX/Linux system. Other privileged virtual machines can also act as message routers.
- A version of Netview is available for z/VM, but it is not widely used.

Some Other Good Stuff To Know

SPOOL – Not just an acronym

- Bonus points if you know what it stands for!
- All virtual machines are normally defined with
  - virtual card reader
  - virtual card punch
  - virtual printer
- They are used to move data and files between virtual machines, and to other systems. This is how TSO XMIT and RECEIVE commands started (and why they use 80-byte records).
- Files created by virtual card punch and printer are stored in spool until either sent to a real printer, or “read” into another virtual machine using virtual reader.
- CP maintains **separate** reader, punch and print queues for all virtual machines.

SPOOL – Not just an acronym

- Examples of some uses:
  - Consoles of Linux and CMS guests can be “spooled” to reader queue of archive virtual machine. This machine reads in each file and stores them on disk for later reference.
  - Email arrives in system spool and is directed to the reader queue of recipient. User runs software in their virtual machine that reads in email from queue and processes it. (Remember PROFS????)
  - Apply maintenance to CMS and then regenerate it. This “punches” a deck of card image to the reader queue, which is then IPLed in order to write the updated system to disk. Yes, you can IPL from the reader! **(Very useful for installing Linux.)**
- SPOOL in VM is not used just for output :
  - System dumps
  - Discontiguous Saved Segments
  - Named Saved Systems
  - System traces
  - National Languages

Spooling Around - RSCS

- Remote Spooling Communication Subsystem (Impress your friends!)
- Purpose is similar to JES2/3 RJE/NJE.
- Allows data and “jobs” to be shipped to remote systems.
  - Output to a real printer connected to the network
  - A job stream to be sent to another system, such as z/OS or VSE
  - A message to users on another RSCS node (was the first form of “chat”)
- Remember VNET? That was IBM’s RSCS network.
- Remember BITNET? Predecessor to the Internet – was also an RSCS network.
- RSCS can talk to JES2/3 over NJE connections.
- It also talks TCP/IP to network based printers and applications.

Don’t Forget the Backups

- CMS based files (minidisk, SFS and BFS) are backed up at file level
  - Allows individual file restore as well as full disk restore
- Non-CMS formatted minidisks must be backed up at track level. Individual file restore not possible.
  - DB2
  - VSAM datasets
  - SFS servers
  - Guest disk volumes such as MVS and Linux guests
    - File-level backups should be done from within guest system
- SPOOL files must be backed up regularly to prevent loss if it becomes corrupted.
  - This is much more important than on MVS because more than jobs and output files are held in SPOOL.

Don’t Forget the Backups

- Native VM backup function is DDR, not sufficient for preserving large disk farms. Good for basic disaster recovery backup.
- CA product, VM:BACKUP is standard for backup/restore of CMS minidisks, SFS directories and non-CMS disks.
- Flashcopy can also be used to create snap backups of large disk volumes if you have Sharks or DS8000’s. PPRC is also supported by z/VM.
- CP command, SPXTAPE can be used to back up SPOOL files to tape.
- CA products VSEG/PLUS and VM:SPOOL can also be used for more advanced SPOOL backup and restore.

Make sure you have your towel…

- z/VM can be intimidating at first….it is so different from other operating systems.
- Its concepts are deceptively simple……all the best ideas are.
- Remember, our mascot is the teddy bear….
- There are always lots of people willing to help.
- IBMVM@LISTSERV.UARK.EDU
- SHARE
- www.vm.ibm.com
- Local VM user groups….see previous link.