The Basics of Using z/VM and CMS

Or, a Day in the Life of an IBM z/VM Virtual Machine

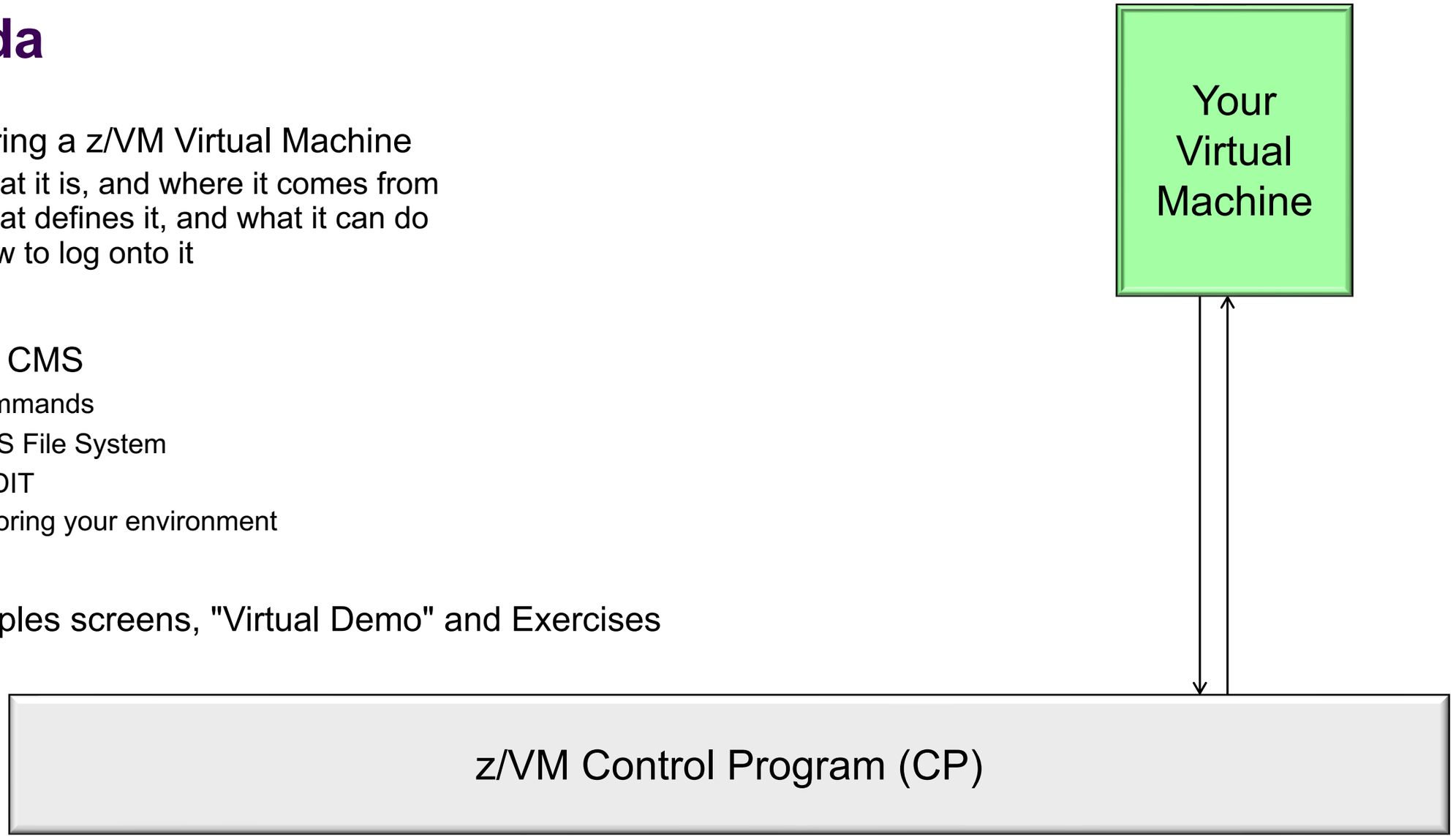
Brian W. Hugenbruch, CISSP
IBM zSystems Security for Virtualization and Cloud
bwhugen@us.ibm.com

V7.2.f – last updated 13 May 2022

#IBMz #zVM

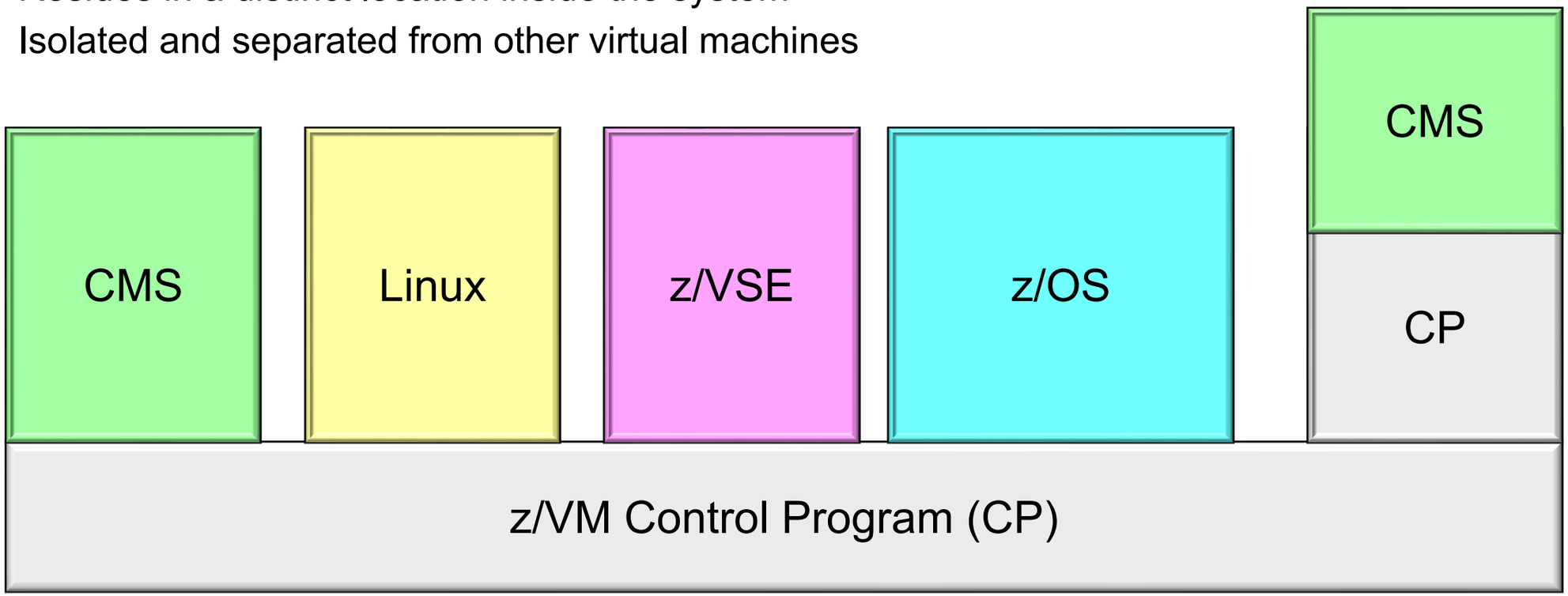
Agenda

- Exploring a z/VM Virtual Machine
  - What it is, and where it comes from
  - What defines it, and what it can do
  - How to log onto it
- Using CMS
  - Commands
  - CMS File System
  - XEDIT
  - Tailoring your environment
- Examples screens, "Virtual Demo" and Exercises



Overview – Virtual Machines

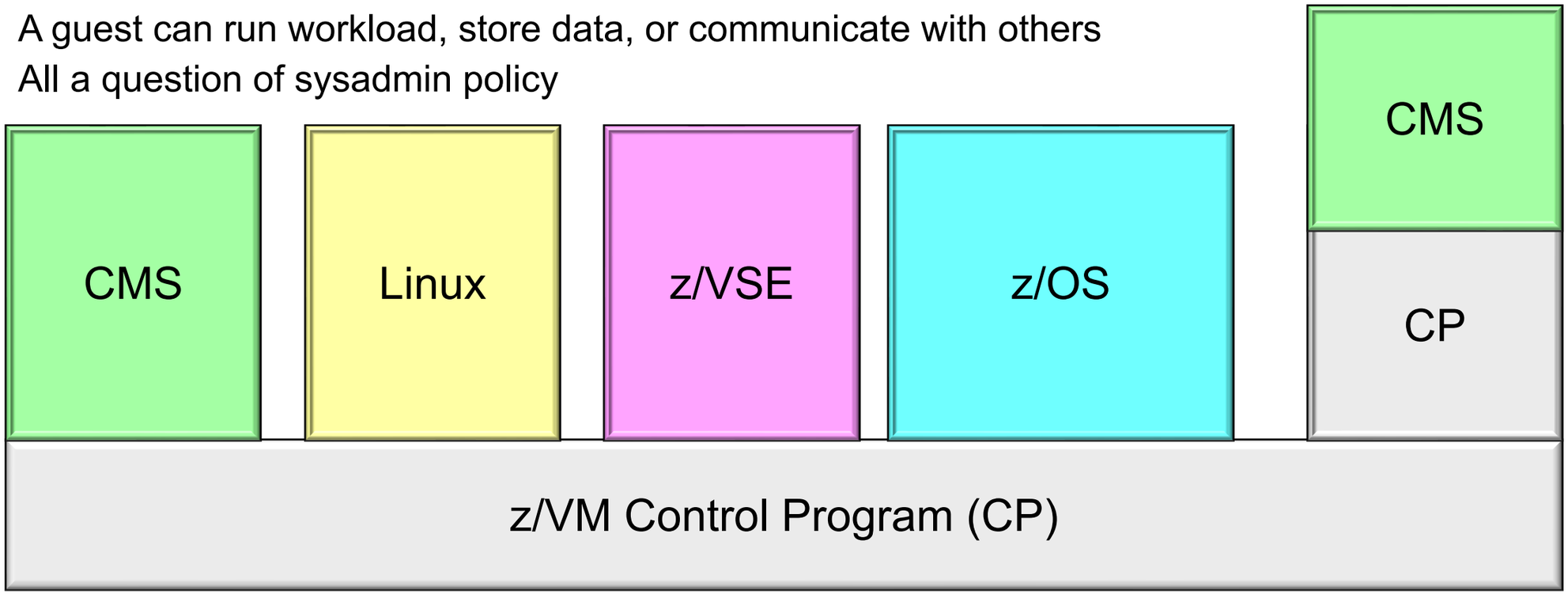
- z/VM contains two basic parts:
  - CP - Control Program (the Hypervisor layer)
  - The guest running under z/VM
    - Resides in a distinct location inside the system
    - Isolated and separated from other virtual machines



#IBMz #zVM

Overview – Virtual Machines

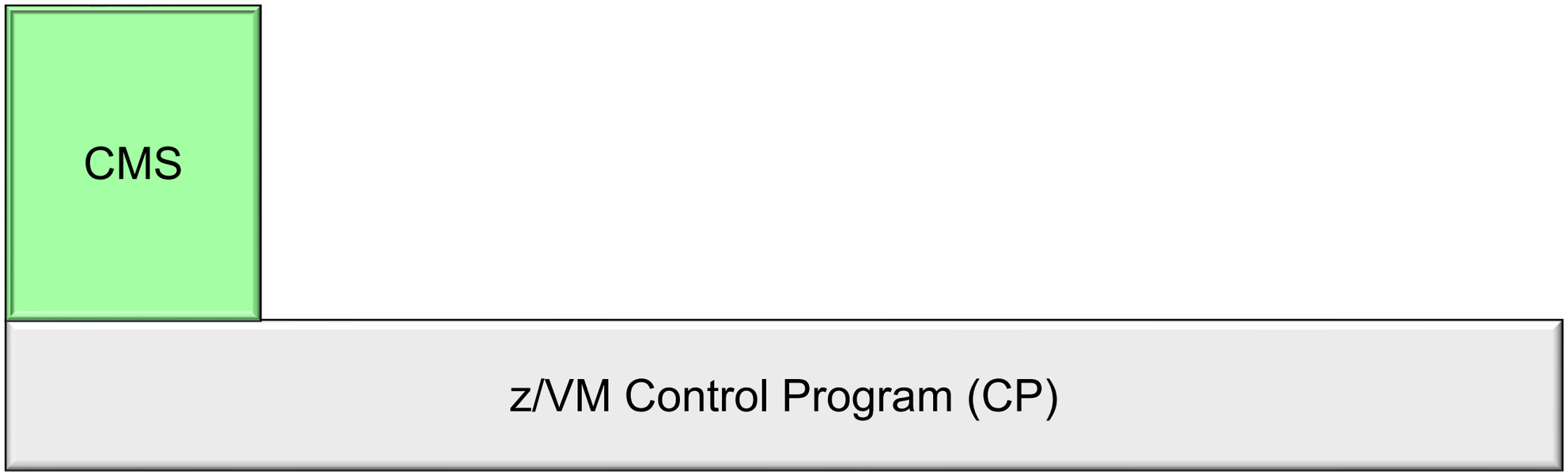
- The virtual machine is:
  - A discrete object on the system, distinct from the hypervisor
  - Isolated from other guests
  - A place where workload runs
    - A guest can run workload, store data, or communicate with others
    - All a question of sysadmin policy



#IBMz #zVM

Overview – Virtual Machines

- **CMS: Conversational Monitor System**
  - An operating system that runs as a guest of VM
  - Provides “a place to stand while you’re configuring the hypervisor”
    - IBM-defined virtual machines with special authorities, for example
  - Can also create files, execute programs, or run workload.



Overview – Hypervisor Parts 101

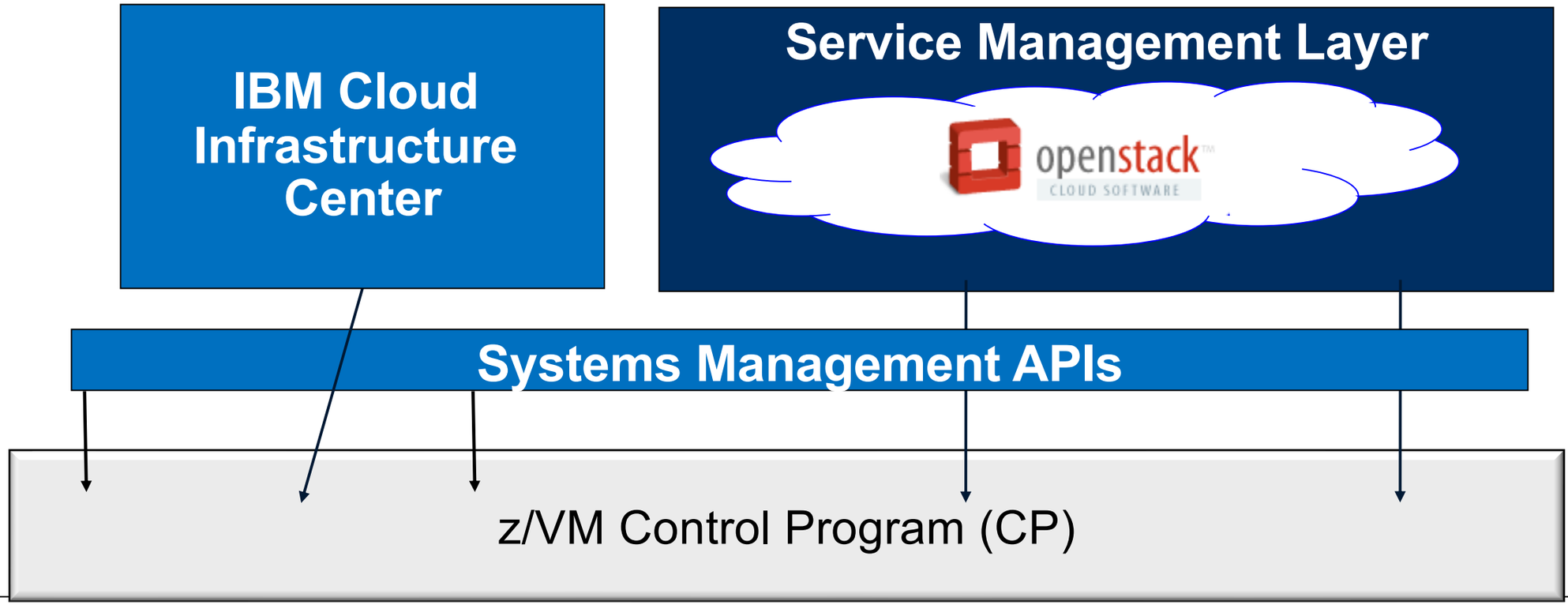
	z/VM	KVM	VMware	PowerVM
Hypervisor	Control Program (CP)	KVM kernel	ESX or ESXi	POWER Hypervisor
Interface	CMS	Virsh or VMM	VirtualCenter	IVM or PowerVC
Guests	CMS, Linux, z/OS, z/VM …	Linux	As advertised	As advertised

Notes:

- CMS is both an **interface to** managing z/VM and a **guest of** z/VM
- The CMS "shell" is not the only way to manage z/VM, but it is a vital one

Overview – Virtual Machines

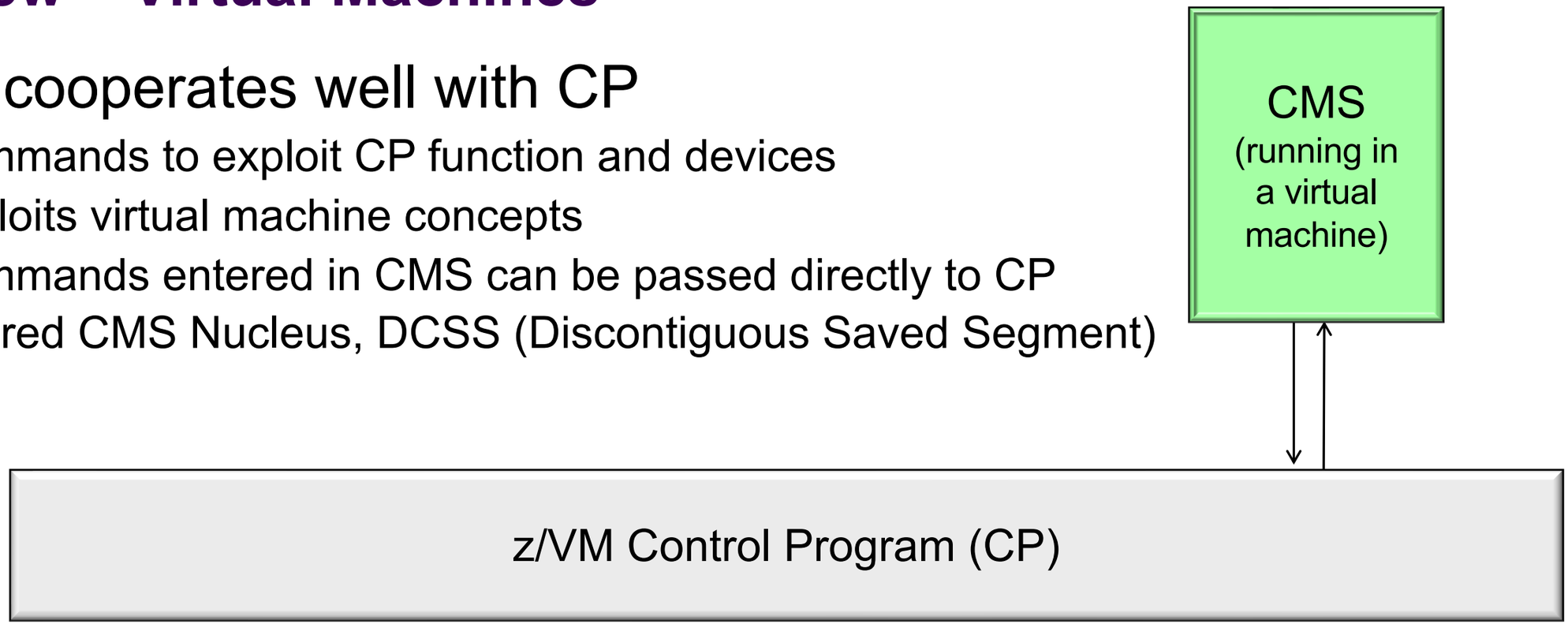
- **Frequently Asked Question:** Is CMS the only way to control my z/VM system?
  - *Answer*: No. Software is available for the management of your systems
  - Managing is relative to your skills and scope of responsibility
- What CMS shows is how z/VM works … no matter how you control it!



#IBMz #zVM

Overview – Virtual Machines

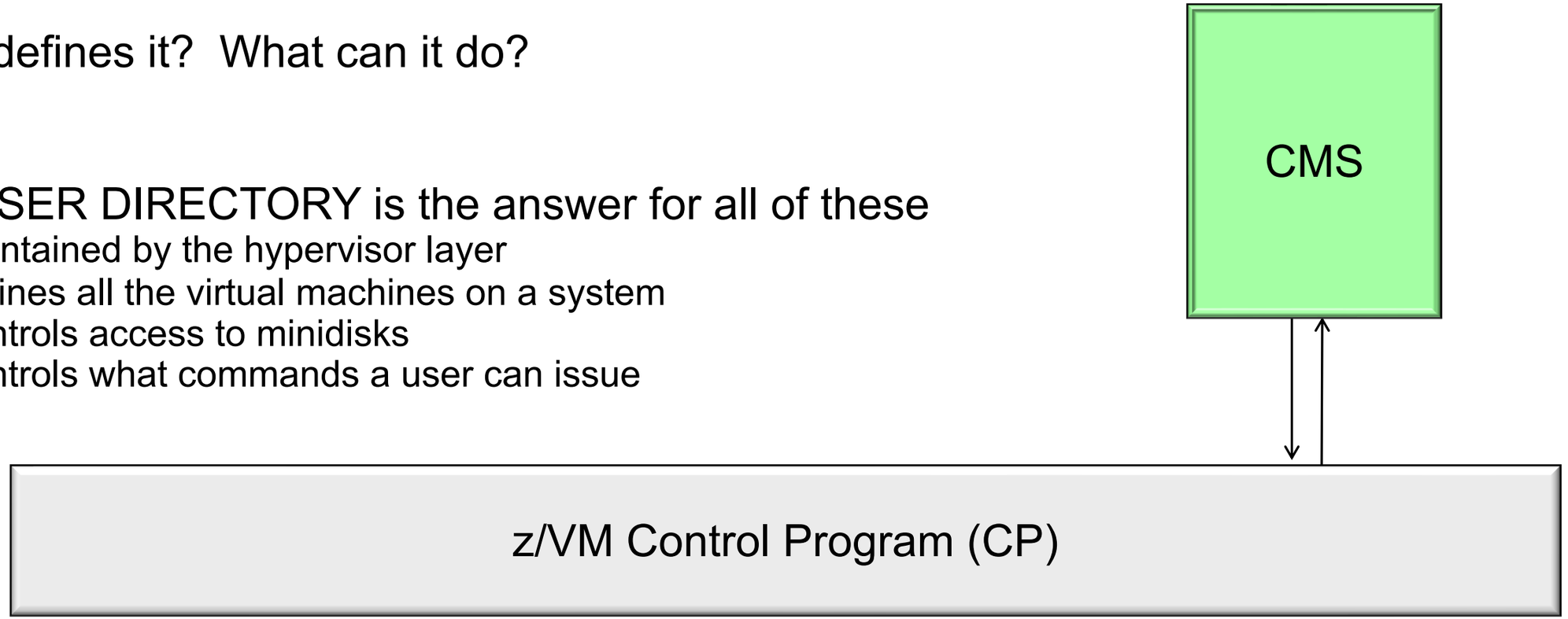
- CMS cooperates well with CP
  - Commands to exploit CP function and devices
  - Exploits virtual machine concepts
  - Commands entered in CMS can be passed directly to CP
  - Shared CMS Nucleus, DCSS (Discontiguous Saved Segment)



- Many productivity tools available
  - IBM-supplied tools
  - Vendor programs
  - REXX programming language – design your own

Where do virtual machines come from?

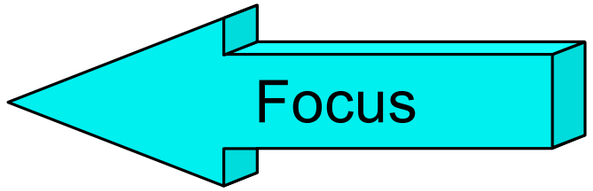
- How does a virtual machine come to be?
- What defines it? What can it do?
- The USER DIRECTORY is the answer for all of these
  - Maintained by the hypervisor layer
  - Defines all the virtual machines on a system
  - Controls access to minidisks
  - Controls what commands a user can issue



Defining a Virtual Machine

Sample User Directory Entry

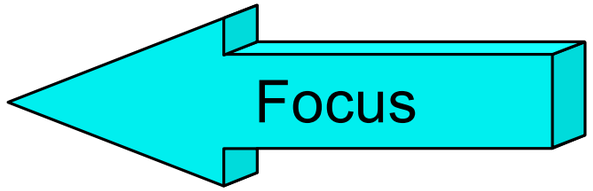
```
USER IBMUSER IBMUSER 32M 32M ABCDEG
 ACCOUNT SYSTEMS
 MACH ESA
 IPL CMS

 CONSOLE 009 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH A
 SPOOL 00E 1403 A

 LINK MAINT 0190 0190 RR * CMS system disk
 LINK MAINT 019E 019E RR * Product code disk
 LINK 7VMRAC20 29E 29E RR
 LINK 7VMRAC20 505 305 RR
 LINK 7VMRAC20 191 391 RR
 MDISK 1191 3390 2078 001 72CRES  MR READ    WRITE    MULTIPLE
 MDISK 191 3390 0895 030 72CUSR  MR READ    WRITE    MULTIPLE
```

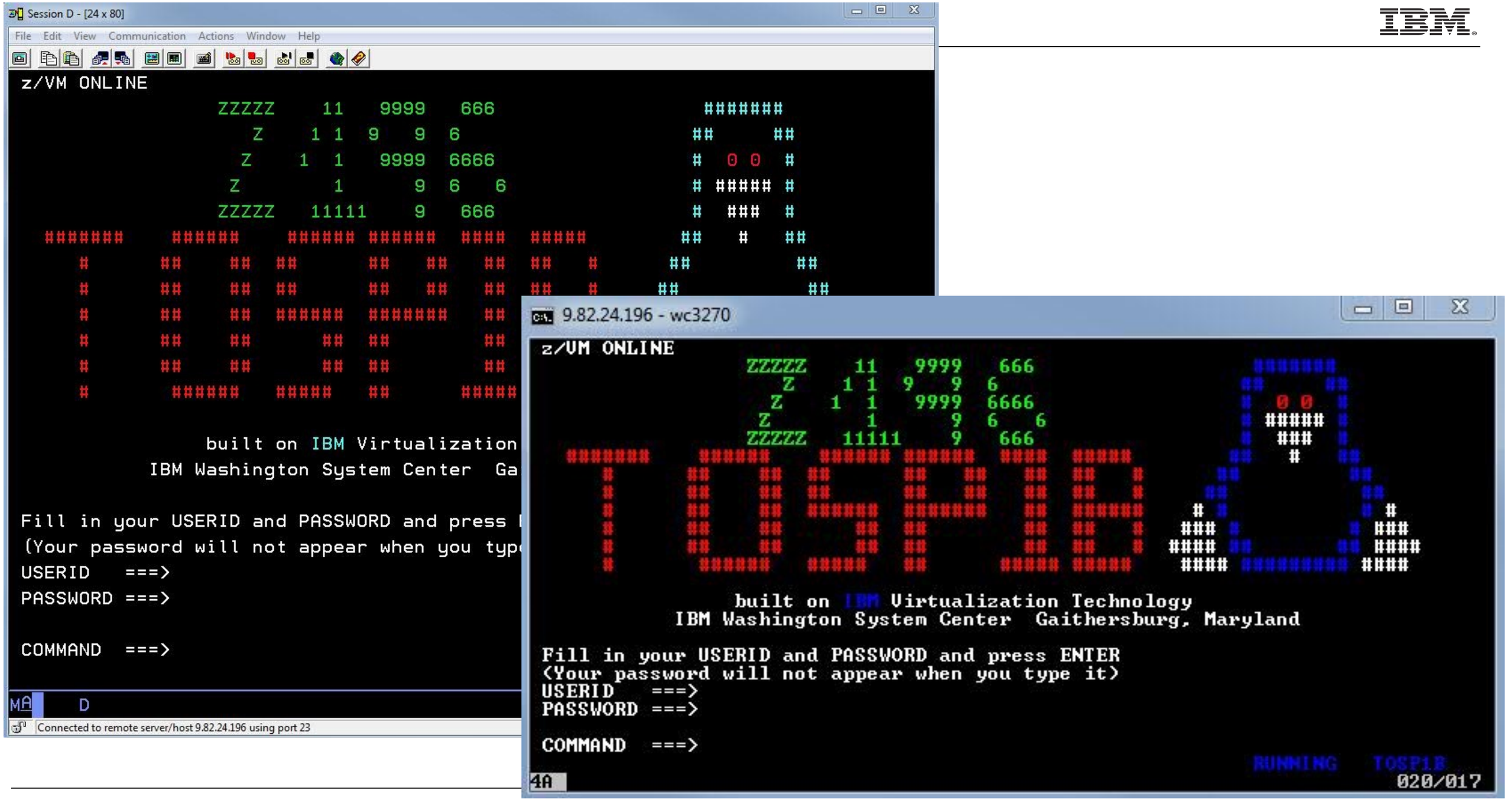


Defining a Virtual Machine

Sample User Directory Entry

```
USER IBMUSER IBMUSER 32M 32M ABCDEG
 ACCOUNT SYSTEMS
 MACH ESA
 IPL CMS

 CONSOLE 009 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH A
 SPOOL 00E 1403 A

 LINK MAINT 0190 0190 RR * CMS system disk
 LINK MAINT 019E 019E RR * Product code disk
 LINK 7VMRAC20 29E 29E RR
 LINK 7VMRAC20 505 305 RR
 LINK 7VMRAC20 191 391 RR
 MDISK 1191 3390 2078 001 72CRES  MR READ    WRITE    MULTIPLE
 MDISK 191 3390 0895 030 72CUSR  MR READ    WRITE    MULTIPLE
```



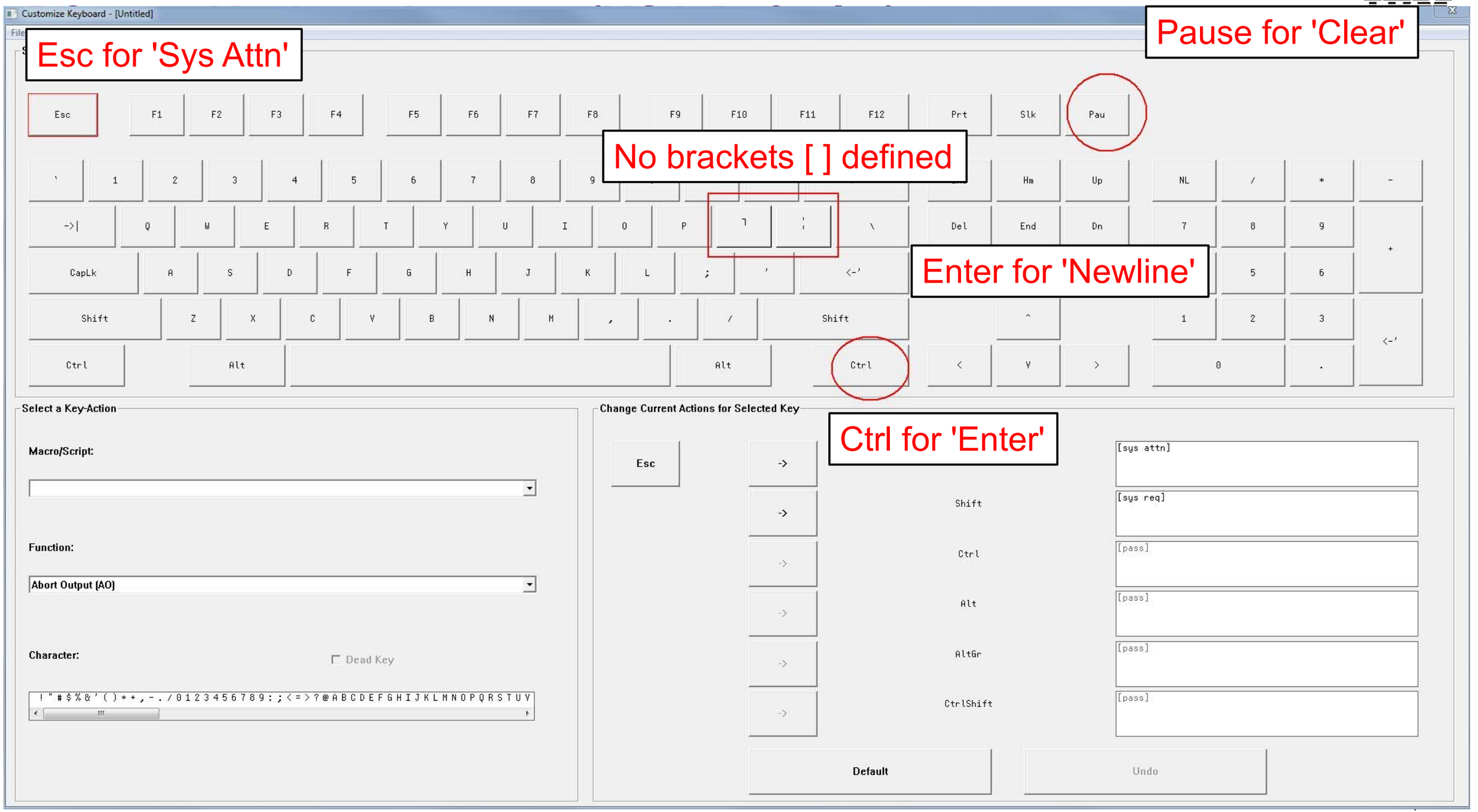


#IBMz #zVM

Overview – CMS

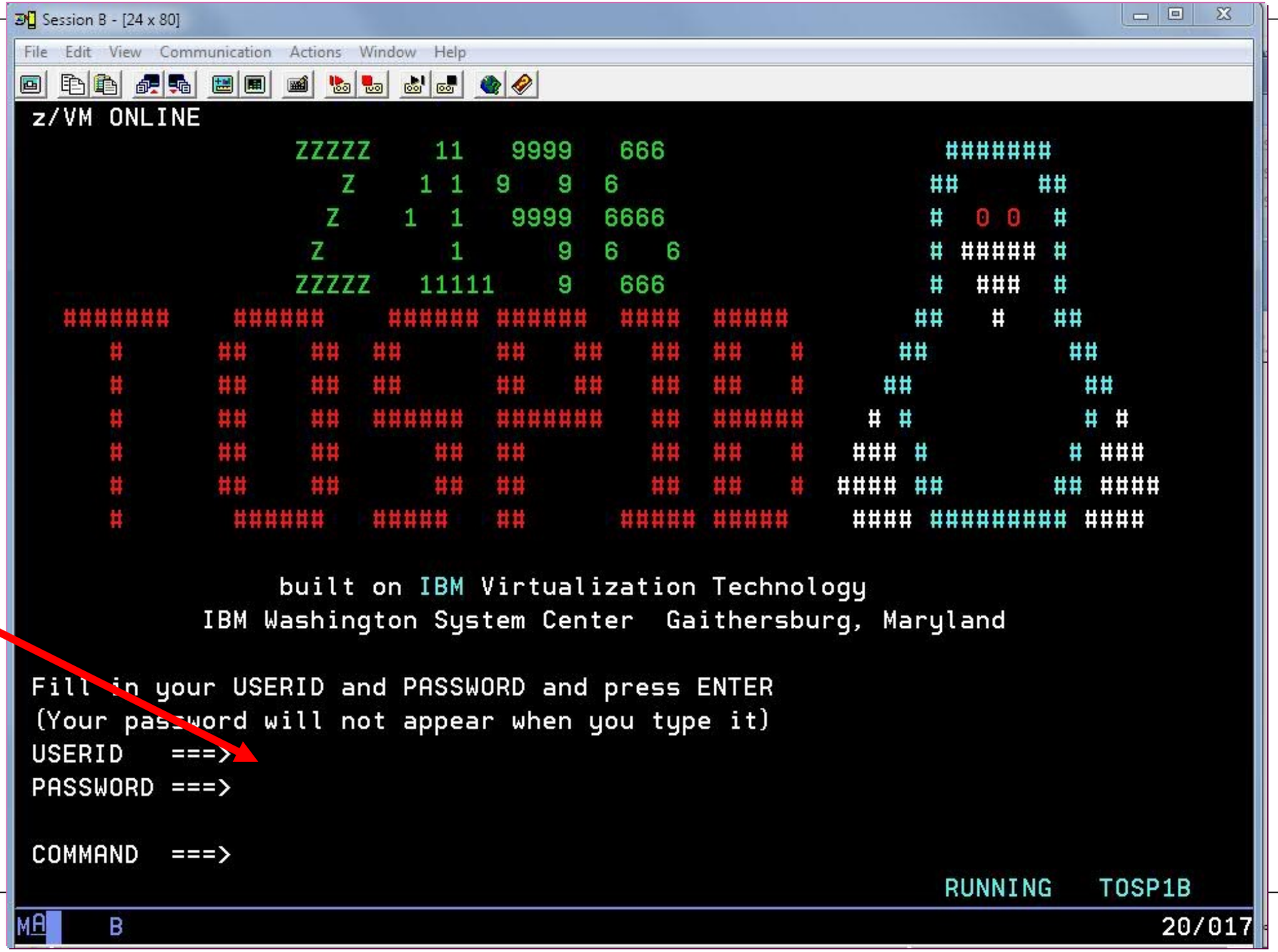
- To establish a z/VM session:

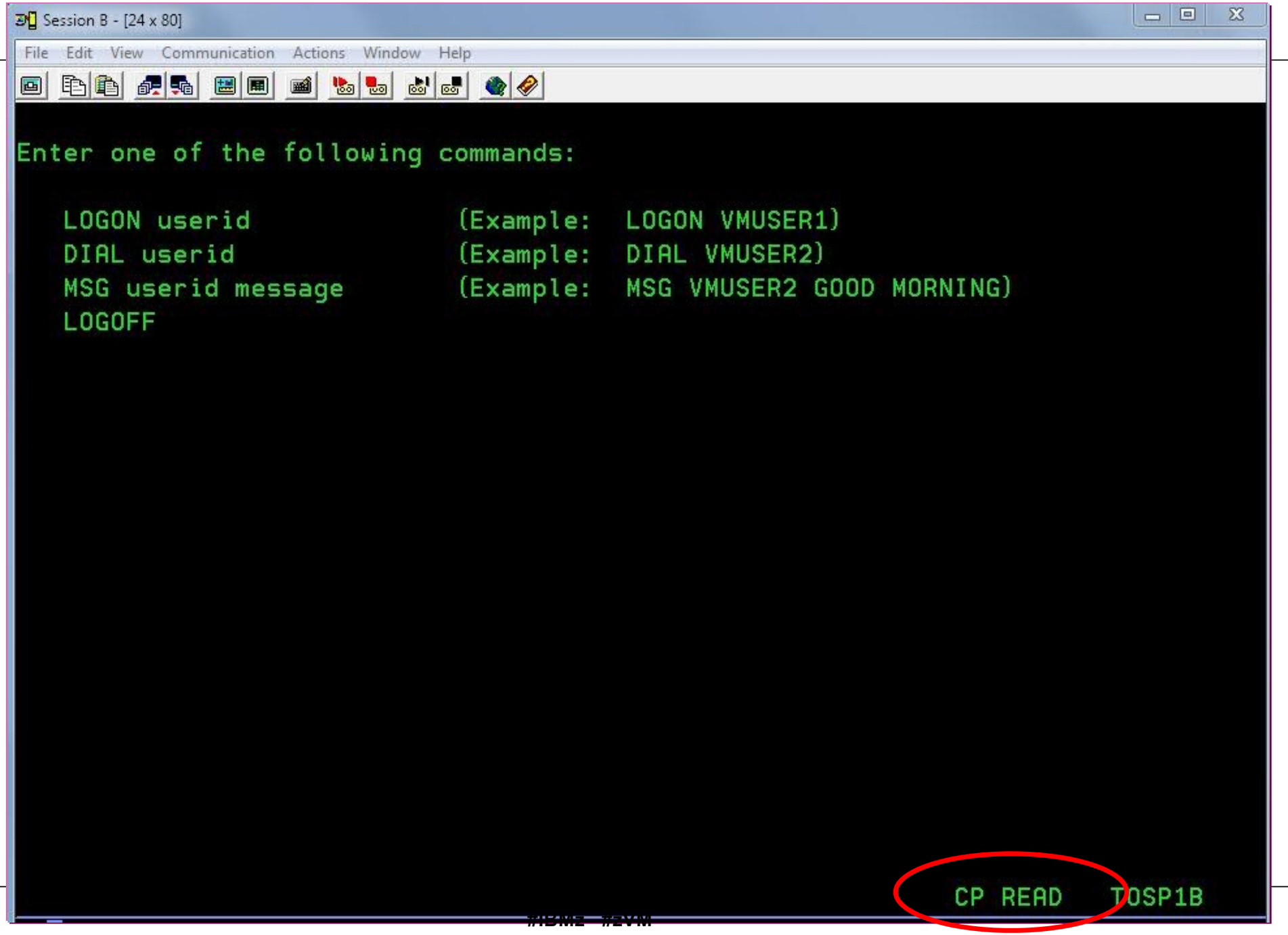
- z/VM Logo Screen
  - One at a time - 3270 emulation (for example, PComm or x3270)
  - **`LOGON <userid> HERE`** -- move a signon to another terminal session
  - **`LOGON <userA> by <userB>`** -- signon using different credentials


Customize Keyboard - [Untitled]
Esc for 'Sys Attn'
Pause for 'Clear'
No brackets [ ] defined
Enter for 'Newline'
Ctrl for 'Enter'
Esc
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
Prt
Slk
Pau
Hm
Up
NL
Del
End
Dn
CapLk
Shift
Ctrl
Alt
Select a Key-Action
Macro/Script:
Function:
Abort Output (AO)
Character:
Dead Key
Change Current Actions for Selected Key
Shift
Ctrl
Alt
AltGr
CtrlShift
[sys attn]
[sys req]
[pass]
Default
Undo

x3270 Keyboard Profile

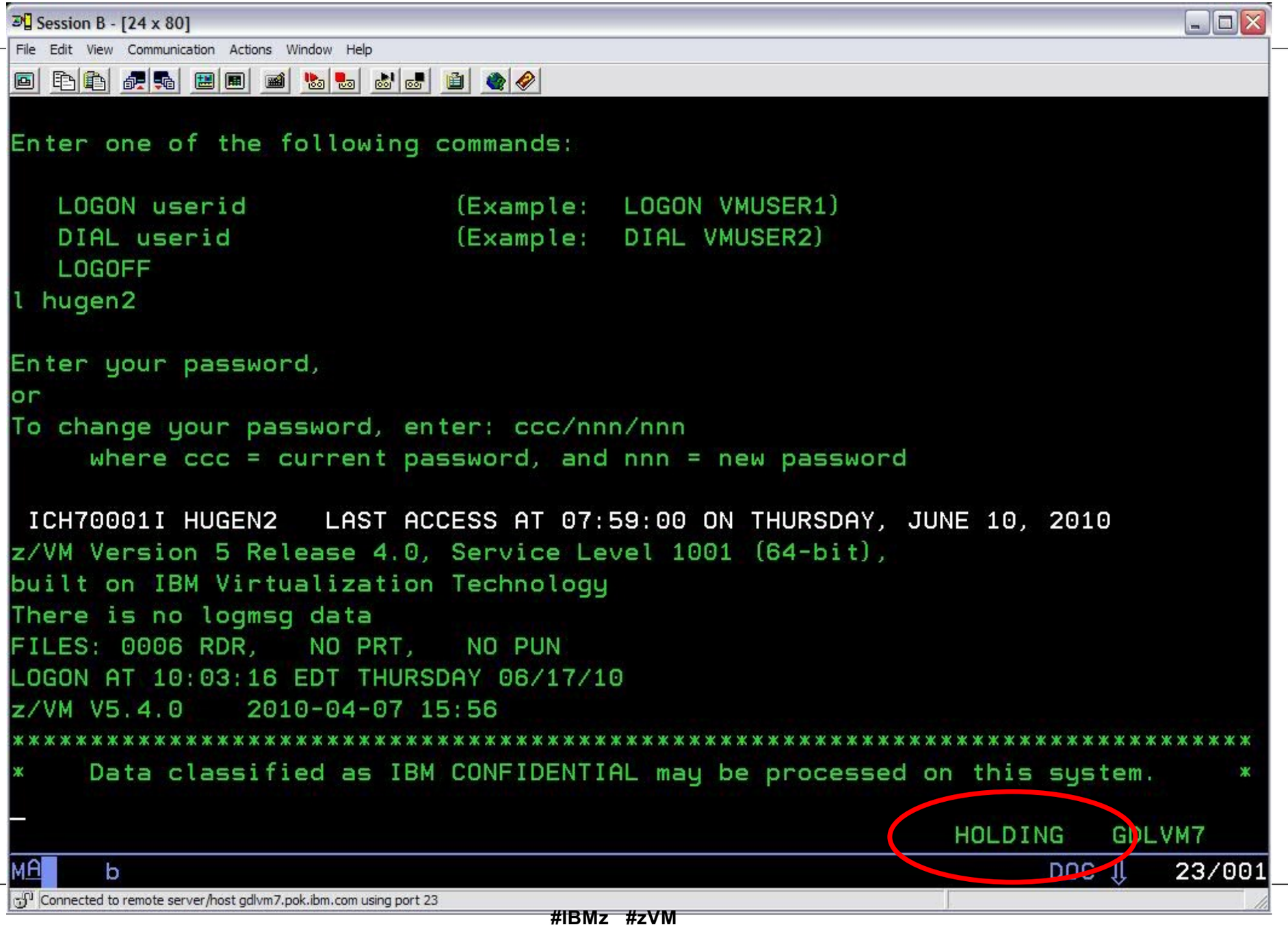
```
! x3270 profile
! File created Wed Mar  9 08:37:27 2011 by x3270 v3.3.10ga4 Thu May 13 10:00:03 UTC 2010 buildd
! This file overrides xrdb and .Xdefaults.
! To skip reading this file, set NOX3270PRO in the environment.
!
! emulator font  (-efont)
x3270.emulatorFont: 3270-20
! model  (-model)
x3270.model: 3
! oversize  (-oversize)
x3270.oversize: 150x60
! Ignore prefix when pasting
x3270.marginedPaste: true
x3270.keymap: default
x3270.keymap.default: \
<Key>Pause:       PA(1)\n\
<Key>Escape:      Clear()\n\
<Key>Insert:      ToggleInsert()\n\
<Key>Return:      Enter()\n\
<Key>Control_R:   Enter()\n\
Ctrl<Key>End:     EraseEOF()\n\
<Key>End:         FieldEnd()\n\
<Key>Home:        BackTab()\n\
Shift<Key>6:      CircumNot()\n\
Shift<Key>Prior:  PF(19)\n\
Shift<Key>Next:   PF(20)\n\
<Btn4Down>:       PF(7)\n\
<Btn5Down>:       PF(8)\n\
<Key>Prior:       PF(7)\n\
<Key>Next:        PF(8)\
```



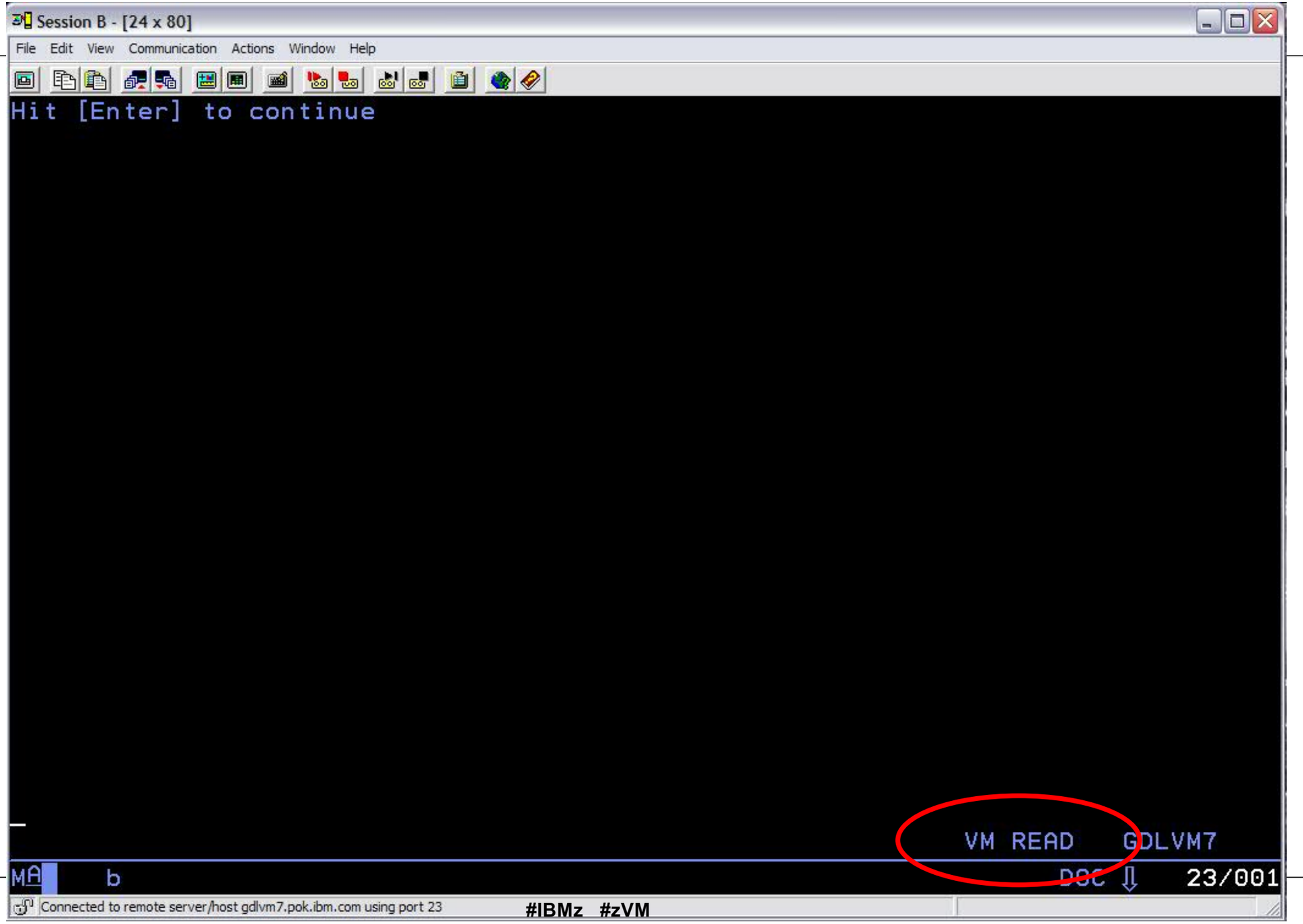


Overview – CMS Execution Modes

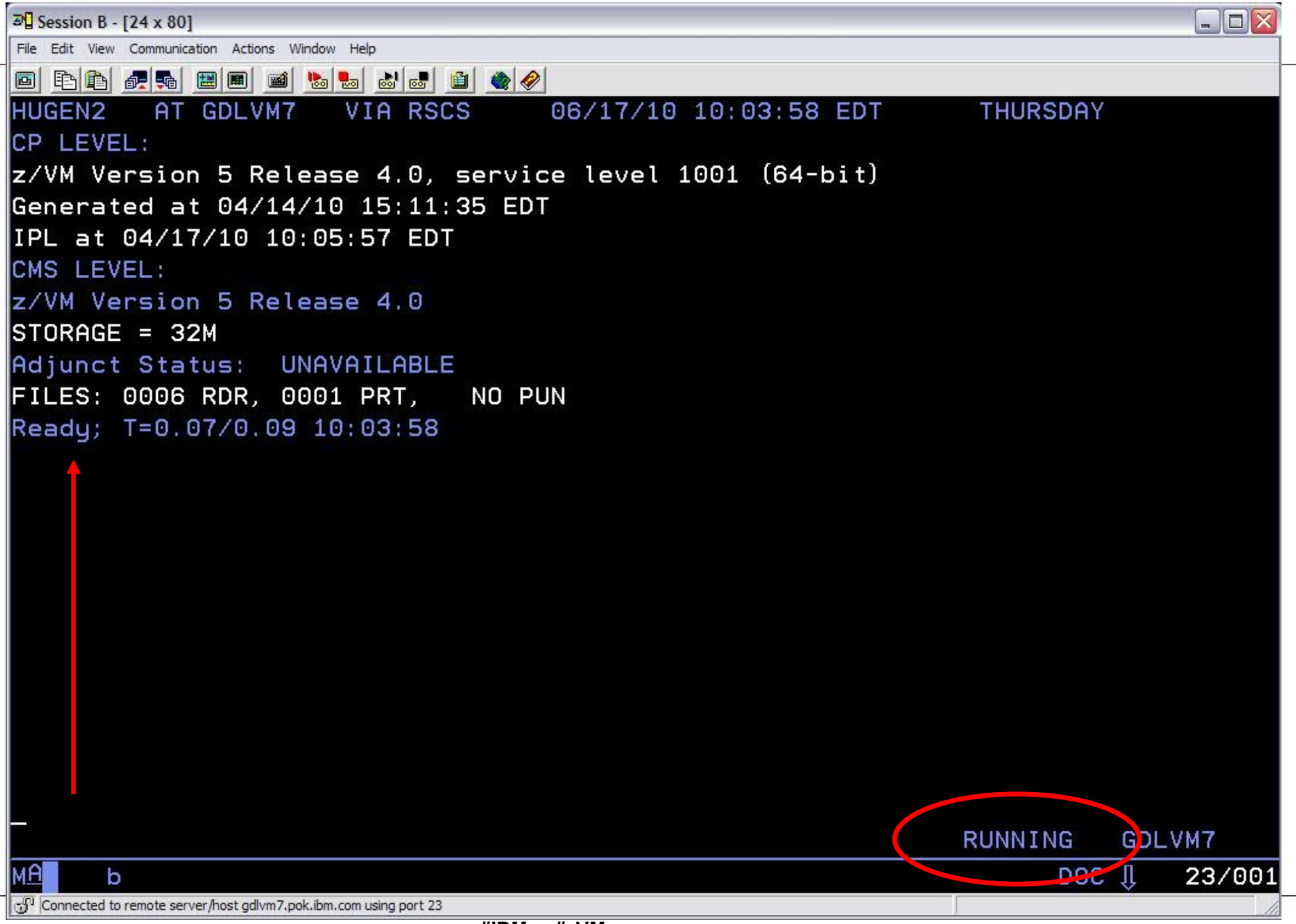
- Status appears at the bottom right of screen
  - **`CP Read`** (CP is waiting for a command)
  - **`VM Read`** (CMS is waiting for a command)
  - **`Running`** (Ready for cmds or working on some)
  - **`More...`** (More info than can fit on the screen)
    - 50 seconds then beep then 10 seconds
    - Determine setting: Query Term
    - Page without waiting: Term More 0 0
  - **`Holding`** (Waiting for you to clear the screen)
  - **`Not Accepted`** (Too many commands in buffer; wait for executing command to complete)



#IBMz #zVM



#IBMz #zVM

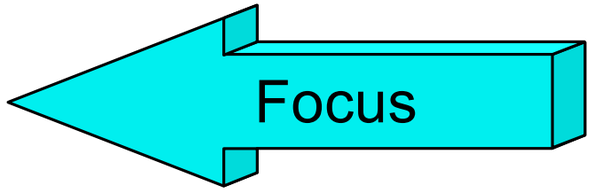


#IBMz #zVM

Defining a Virtual Machine

Sample User Directory Entry

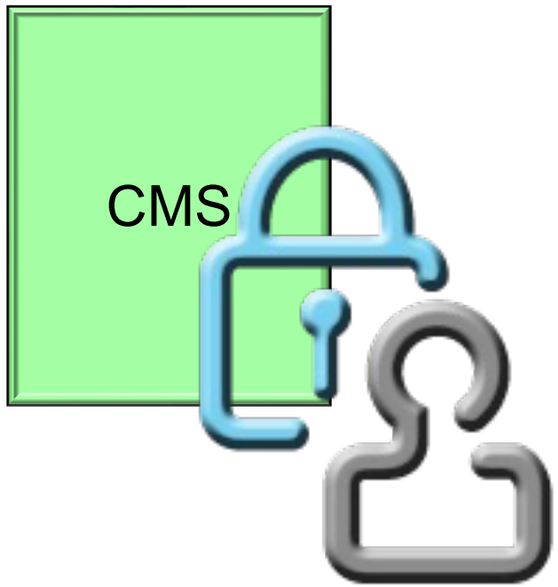
```
USER IBMUSER IBMUSER 32M 32M ABCDEG
 ACCOUNT SYSTEMS
 MACH ESA
 IPL CMS

 CONSOLE 009 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH A
 SPOOL 00E 1403 A

 LINK MAINT 0190 0190 RR * CMS system disk
 LINK MAINT 019E 019E RR * Product code disk
 LINK 7VMRAC20 29E 29E RR
 LINK 7VMRAC20 505 305 RR
 LINK 7VMRAC20 191 391 RR
 MDISK 1191 3390 2078 001 72CRES  MR READ    WRITE    MULTIPLE
 MDISK 191 3390 0895 030 72CUSR  MR READ    WRITE    MULTIPLE
```



Defending a Single Virtual Machine

There are seven IBM-defined Privilege Classes:

```
   A:  Commands for System Operators
   B:  Commands for System Resource Operators
   C:  Commands for System Programmers
   D:  Commands for Spooling Operators
   E:  Commands for System Analysts
   F:  Commands for Service Representatives
   G:  Commands for General CMS Users
 ANY:  Commands available to everyone
```

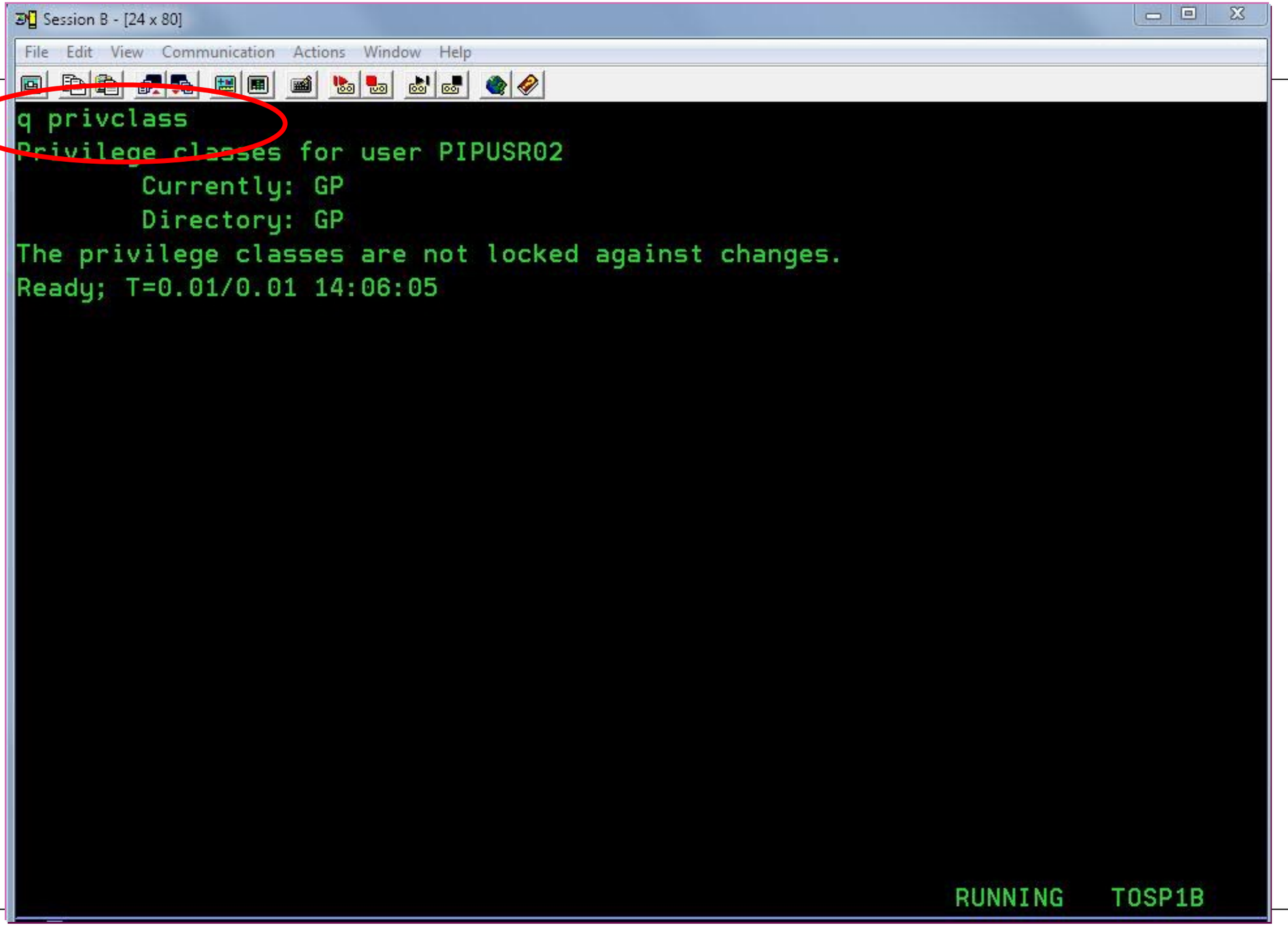


The capabilities of a virtual machine can therefore be defined based upon the role or roles it is expected to carry out (Role-Based Access Control). System administrators can define their own privilege classes

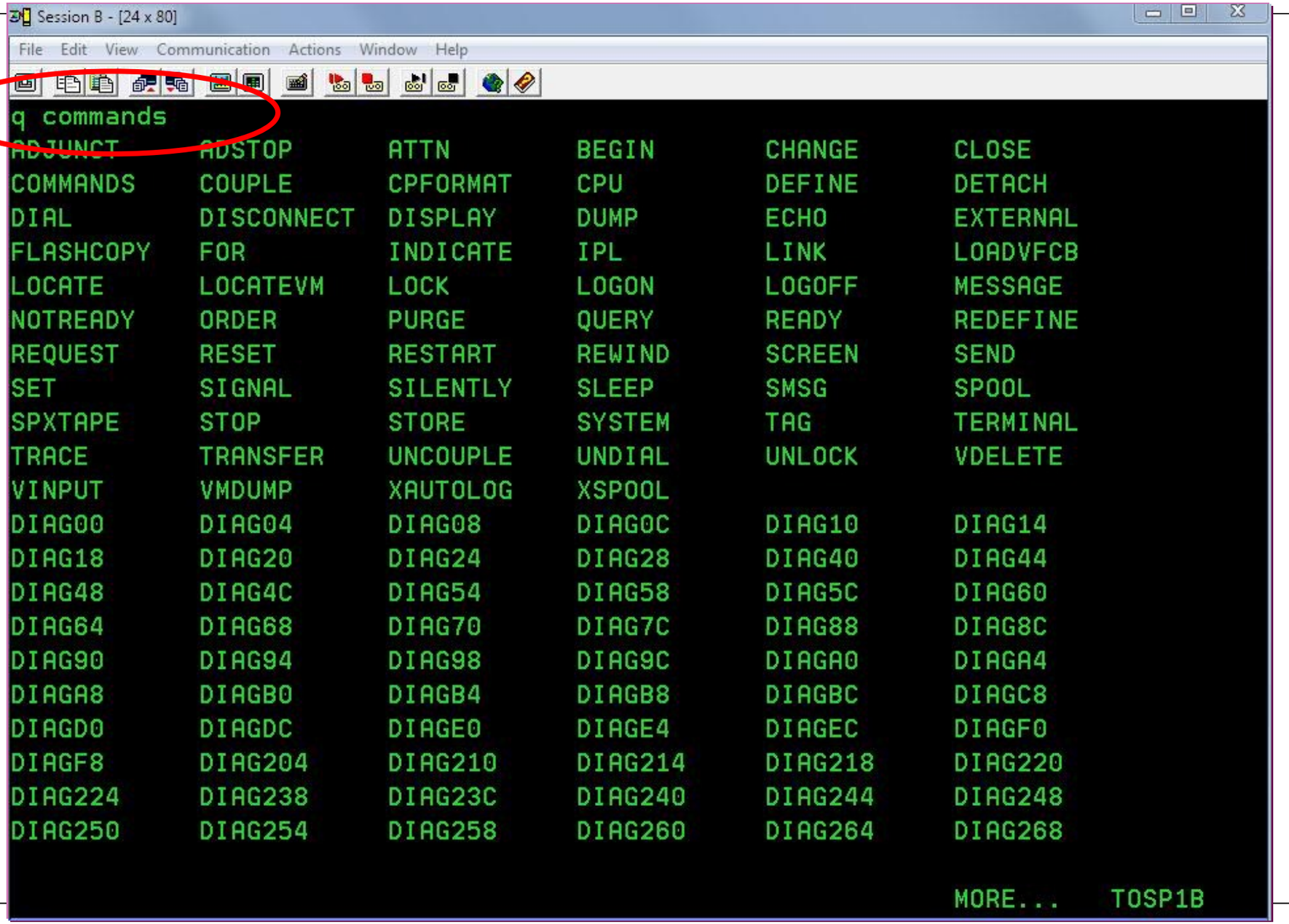
- **QUERY PRIVCLASS**
  - Lists current privilege class for your virtual machine
- **QUERY COMMANDS**
  - Provides a list of all the commands to which your VM is authorized
  - **Note**: a security product may refine security policy on your VM system

Try for yourselves! (5-minute exercise)

- Change your PComm keyboard layouts to something you find useful
- LOGON <userid> /* Remember, CTRL is default Enter key */
  /* ... Until you change it, that is. */
- QUERY PRIVCLASS /* Your security context ... */
- QUERY COMMANDS



#IBMz #zVM

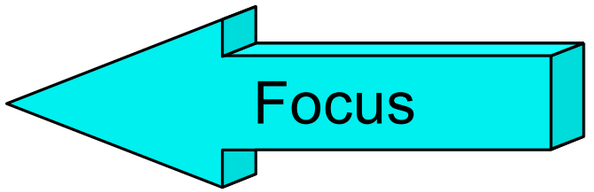


#IBMz #zVM

Defining a Virtual Machine

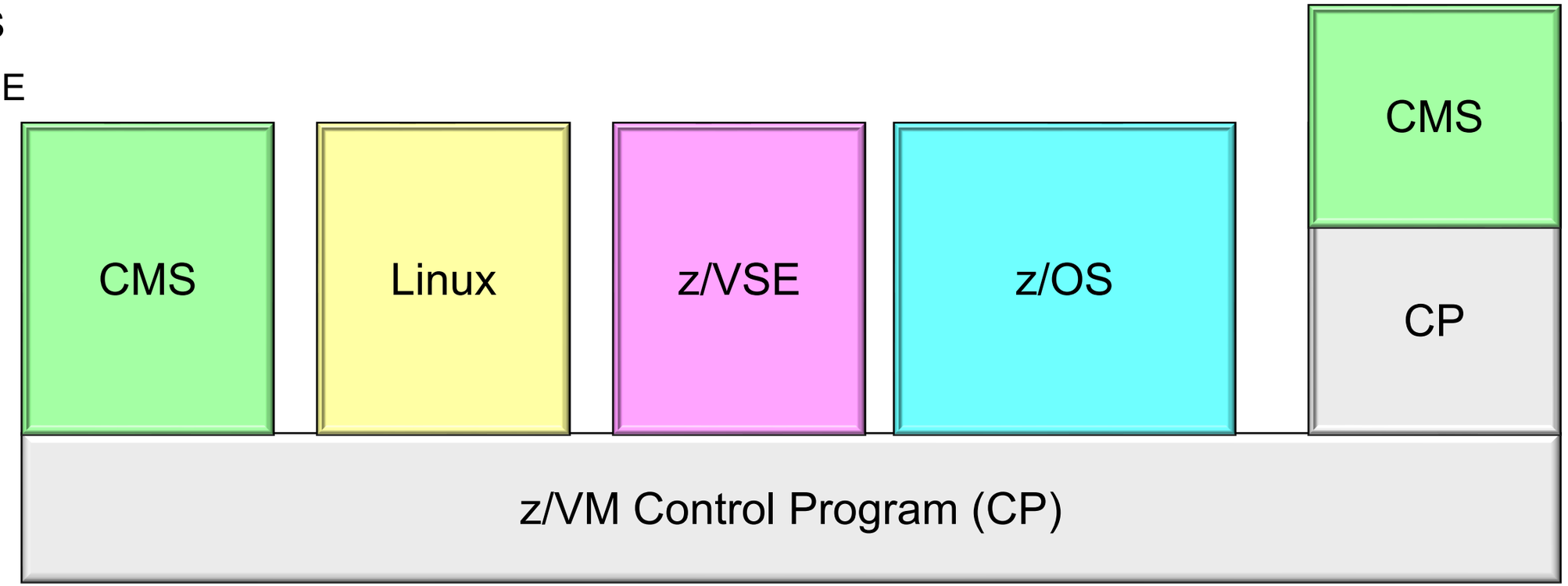
Sample User Directory Entry

```
USER IBMUSER IBMUSER 32M 32M ABCDEG
 ACCOUNT SYSTEMS
 MACH ESA
 IPL CMS

 CONSOLE 009 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH A
 SPOOL 00E 1403 A

 LINK MAINT 0190 0190 RR * CMS system disk
 LINK MAINT 019E 019E RR * Product code disk
 LINK 7VMRAC20 29E 29E RR
 LINK 7VMRAC20 505 305 RR
 LINK 7VMRAC20 191 391 RR
 MDISK 1191 3390 2078 001 72CRES  MR READ    WRITE    MULTIPLE
 MDISK 191 3390 0895 030 72CUSR  MR READ    WRITE    MULTIPLE
```



Defining a Virtual Machine
Sample User Directory Entry

- CMS
- zCMS (the 64-bit version of CMS)
- Linux (often on its own separate disk)
- z/OS
- z/VSE



#IBMz #zVM

CMS Commands

- CMS commands control the virtual machine
  - Manipulate disks and files
  - Adjust the operating environment

- Commands are blank-delimited

- Commands are case-insensitive
  - CMS will automatically uppercase and pass to command parser

- General syntax:
  Command name [operand(s)...] [(options.... [ )]]

  Examples:

```
copy Profile Exec A = = C
Rdrlist
LISTFILE (Date
```

CMS Commands

- Command Search Order -- when a command is entered, CMS has to locate it
  - Search for an EXEC with the specified command name
    - EXECs in storage
    - Command name with Filetype EXEC on accessed disk or directory (A-Z)
  - Search for translation or synonym
  - Search for a module with the specified command name
    - Nucleus extension, transient area, nucleus resident, on accessed disk/directory
  - If Command not found in CMS, it will be passed to CP for execution
    - SET IMPCP (Implied CP) can enable or disable this behavior

CMS Commands and CP Commands

- CP (the hypervisor) and CMS (the guest) have their own sets of commands

- Commands can be passed directly to CP, rather than going through the CMS search order
  - Useful if CMS is "stuck" (e.g., NOT ACCEPTED)
  - **`#CP <command> (options`**

- #CP IPL CMS – restarts your entire CMS guest
  - "Reboot button" / "percussive maintenance"
  - Destroys any unsaved work

CMS Commands

- Immediate Commands
  - Can be entered while another command is running
  - Interrupts the running command and executes immediately
  - 10 system immediate commands:
    HB, **HI**, HO, HT**, HX,** RT, RO, SO, TE, TS

    Tip: HX is the equivalent of ^C or ^X in other operating systems
  - Advanced users can define their own immediate commands (IMMCMD)

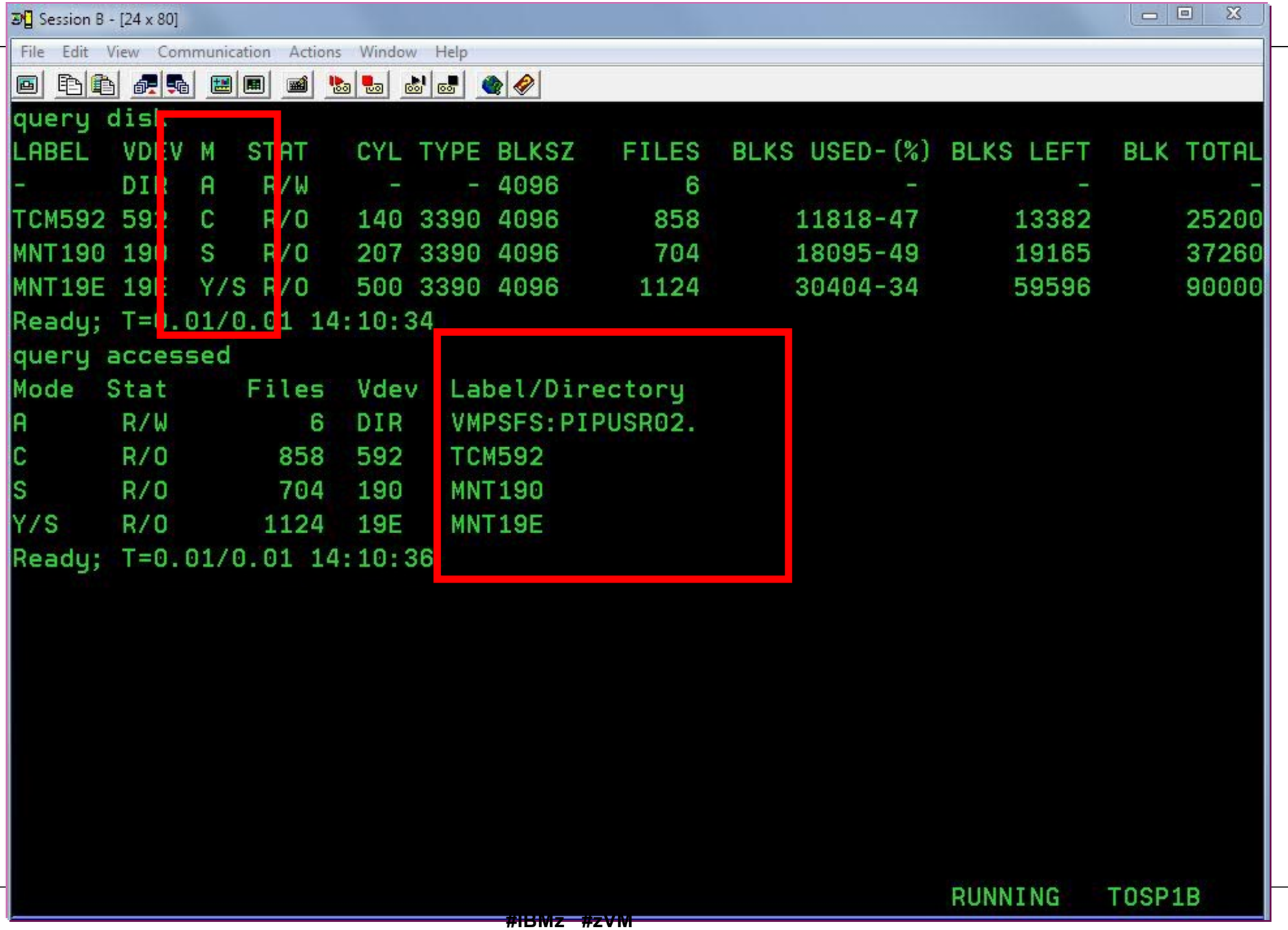
CMS File System

- Files are named using a file identifier (file ID) consisting of 3 fields:
  - File name (FN)
  - File type (FT)
  - File mode (FM) or Directory name (dirname) – **a letter A through Z**.

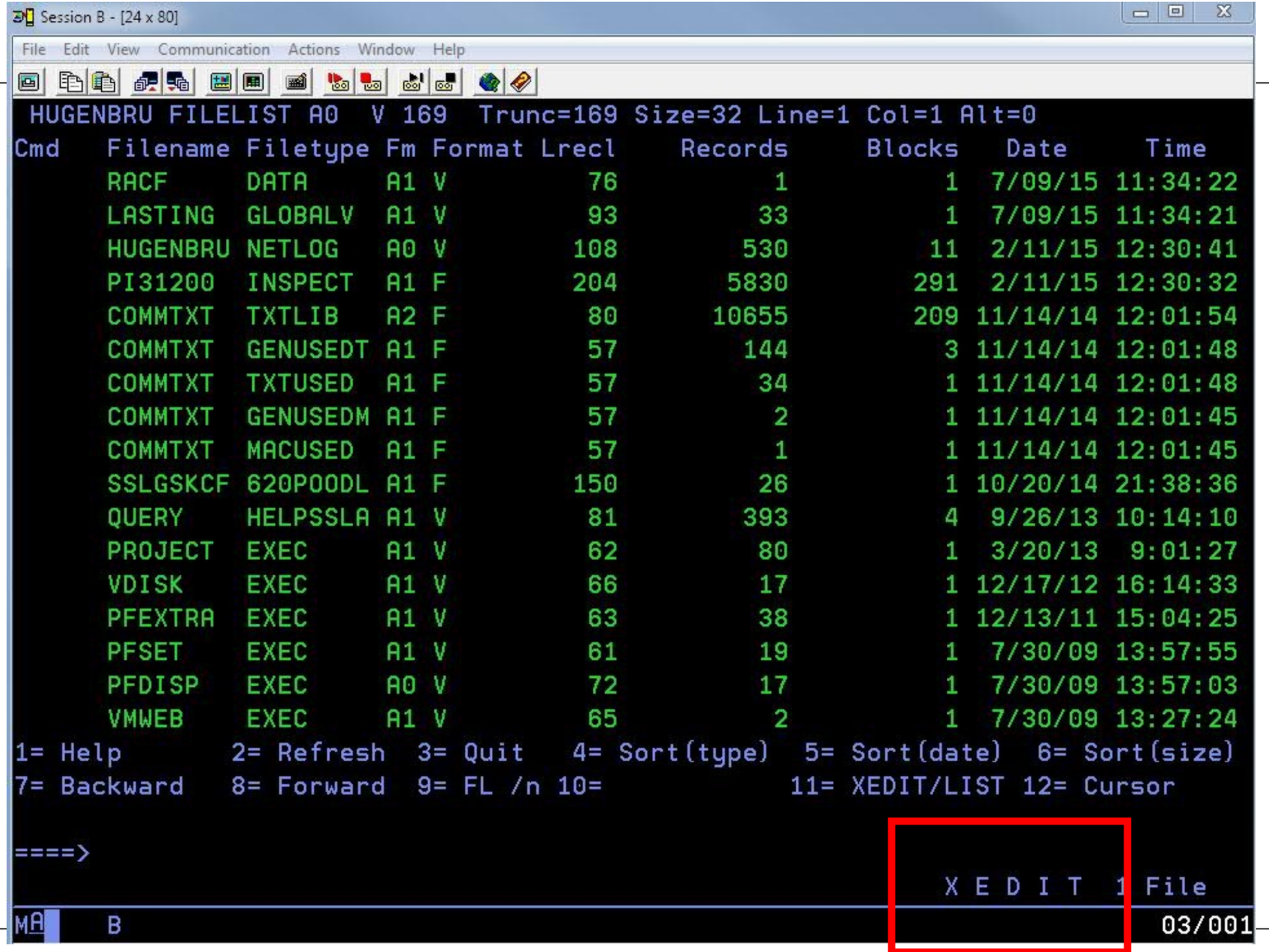
- Files can be stored in a few different ways:
  - On Minidisks (fn ft fm)
    - Standard file modes: A - user's disk, S - system disk
  - In an SFS (Shared File System) filespace: `GPLSRV2:HUGENBRU.REXX.`
  - On the BFS (Byte File System) `/../home/userid/`
    - hierarchical file structure
  - In NFS (Network File System)

Try for yourselves! (8-minute exercise)

```
QUERY DISK                                /* Your resources ... */
QUERY ACCESSED
FILELIST * * A
LISTFILE * * A
SET FILEPOOL GPLSRV2                      /* connect to an SFS */
ACCESS GPLSRV2:HUGENBRU.NEWHIRE. D        /* or another free filemode */
FILELIST * * D                            /* see what's out there */
QUERY AUTH .NEWHIRE                       /* is this different from ... */
QUERY AUTH GPLSRV2:HUGENBRU.NEWHIRE       /* who has access? */
LISTFILE FOREVER * *                      /* and if file exists, then ... */
EXEC FOREVER
/* Now, break out of the EXEC ...                        */
/* Question: are you in RUNNING state? If not ...  */
B
```



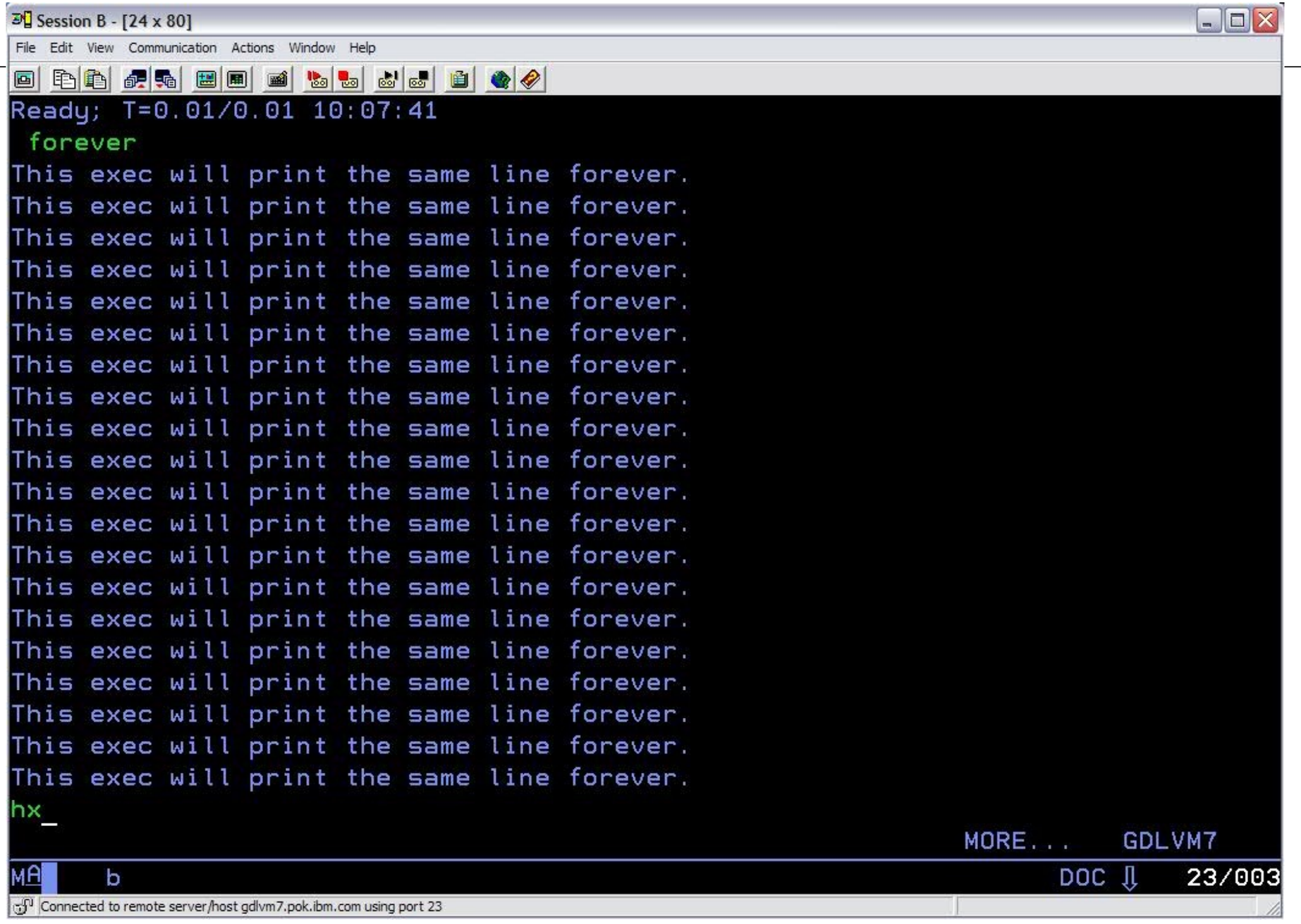
#IBMZ #ZVM



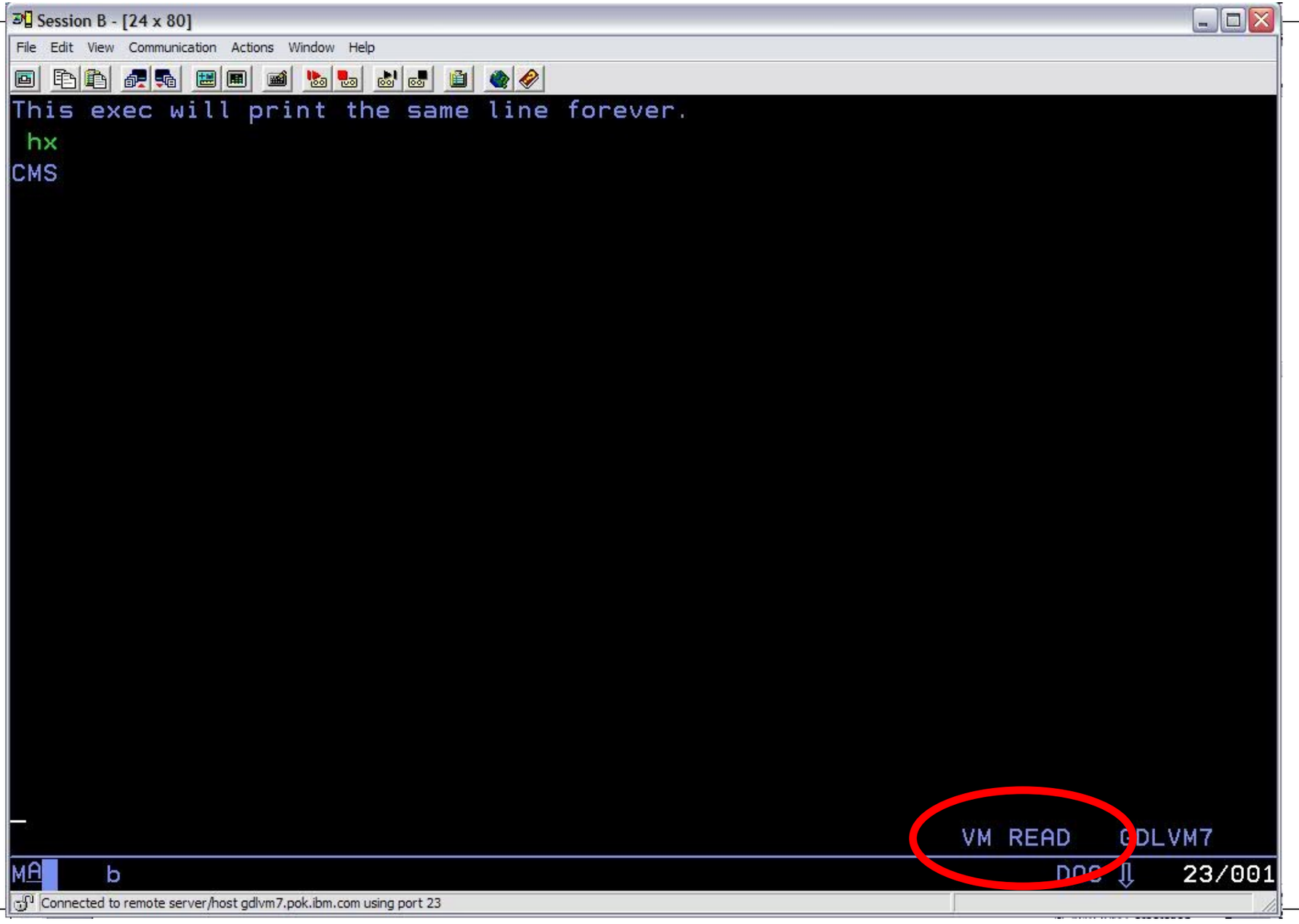
#IBMz #zVM

Overview – CMS

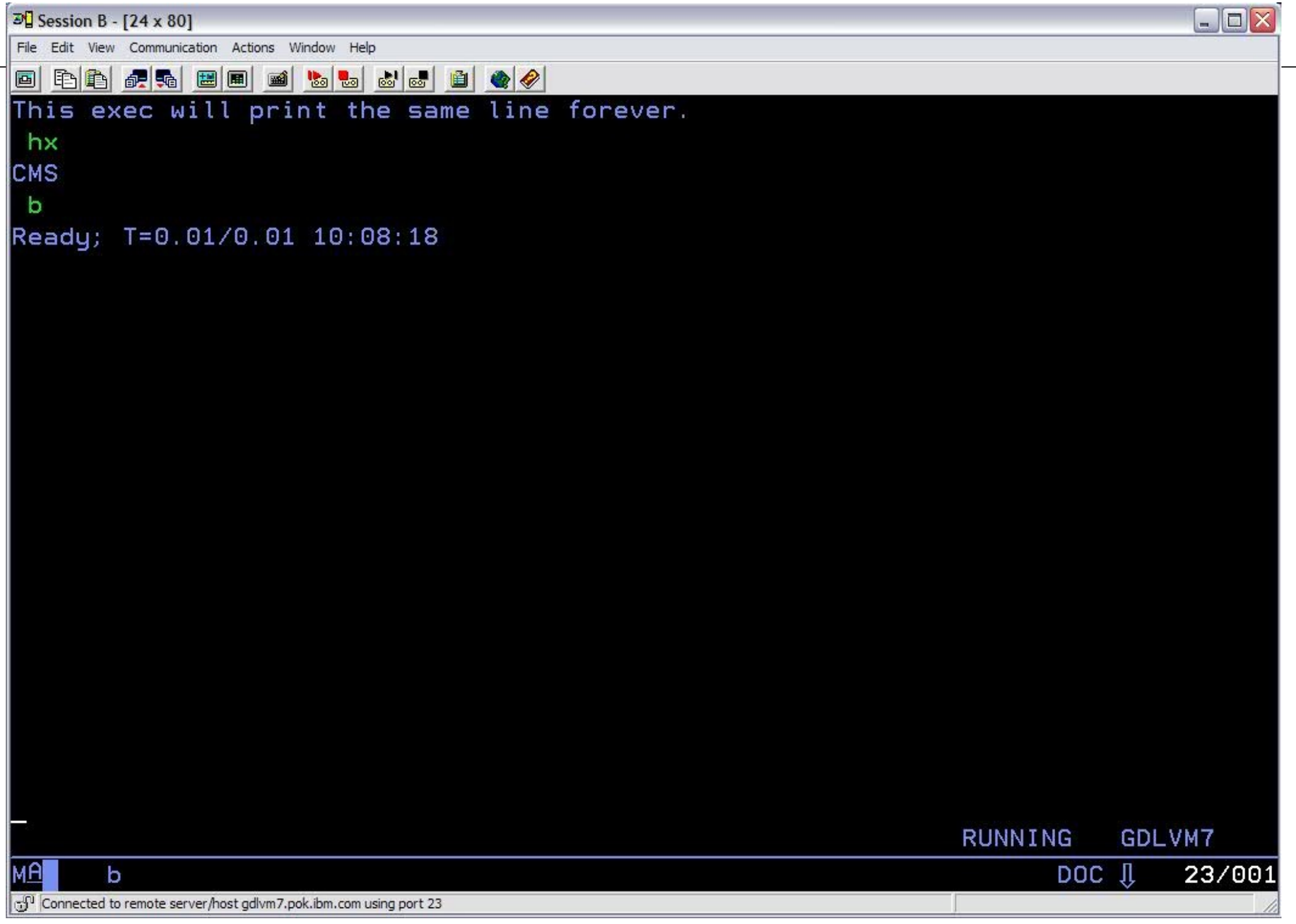
- CMS supports multiple internal environments:
- CMS
  - **`IPL CMS`** or **`Begin`** will run **`Profile Exec`**
  - Linemode or Fullscreen mode
- XEDIT Environment
  - XEDIT ***fn ft fm,***
  - Tailorable **(Profile Xedit)**
  - CMS Subset mode
- Extensions*
  - Open Extensions (Posix Shell & Utilities)
  - Byte File System, Network File System
  - z/OS and z/VSE simulation modes



#IBMz #zVM



#IBMz #zVM

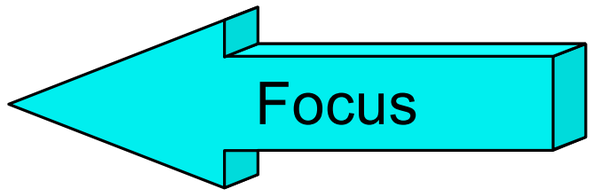


#IBMz #zVM

Defining a Virtual Machine

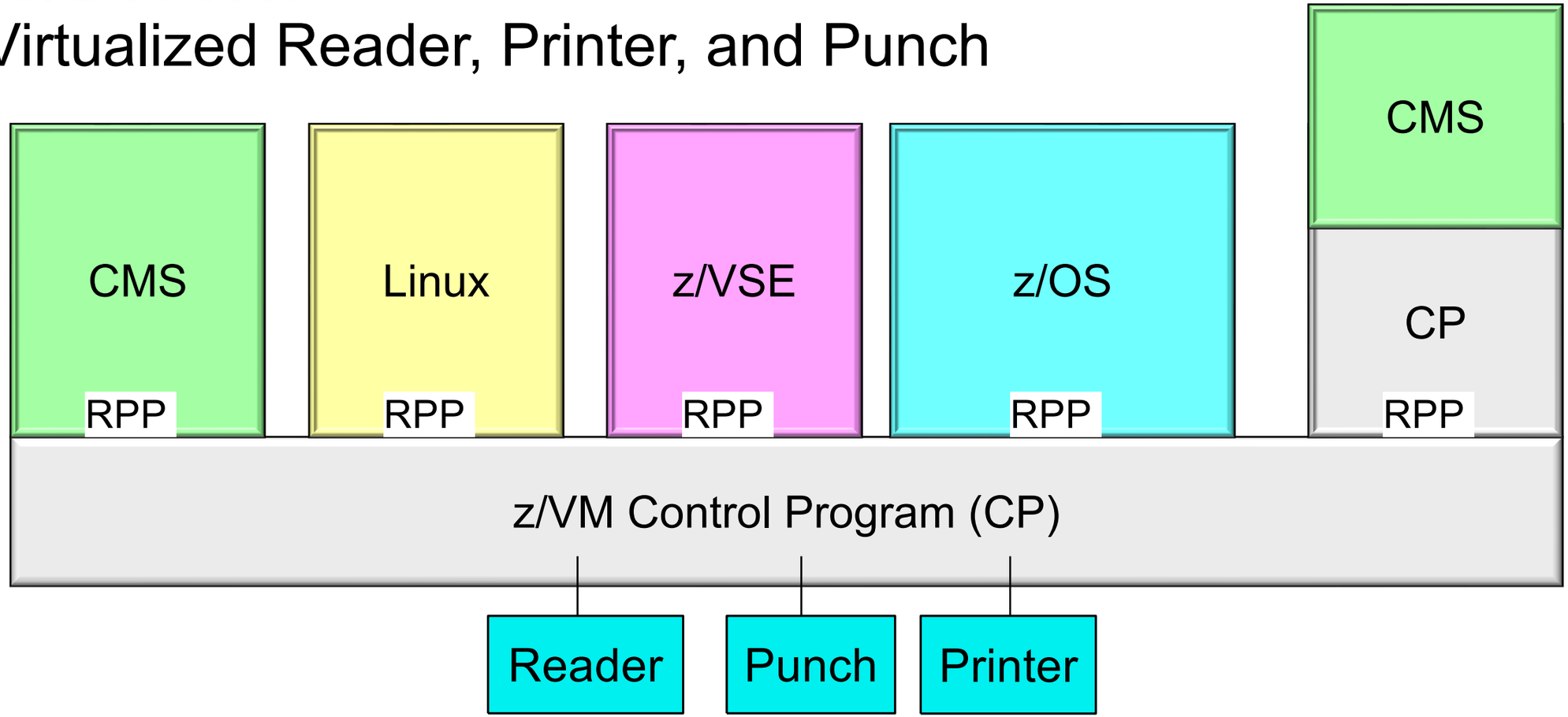
Sample User Directory Entry

```
USER IBMUSER IBMUSER 32M 32M ABCDEG
 ACCOUNT SYSTEMS
 MACH ESA
 IPL CMS

 CONSOLE 009 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH A
 SPOOL 00E 1403 A

 LINK MAINT 0190 0190 RR * CMS system disk
 LINK MAINT 019E 019E RR * Product code disk
 LINK 7VMRAC20 29E 29E RR
 LINK 7VMRAC20 505 305 RR
 LINK 7VMRAC20 191 391 RR
 MDISK 1191 3390 2078 001 72CRES  MR READ    WRITE    MULTIPLE
 MDISK 191 3390 0895 030 72CUSR  MR READ    WRITE    MULTIPLE
```

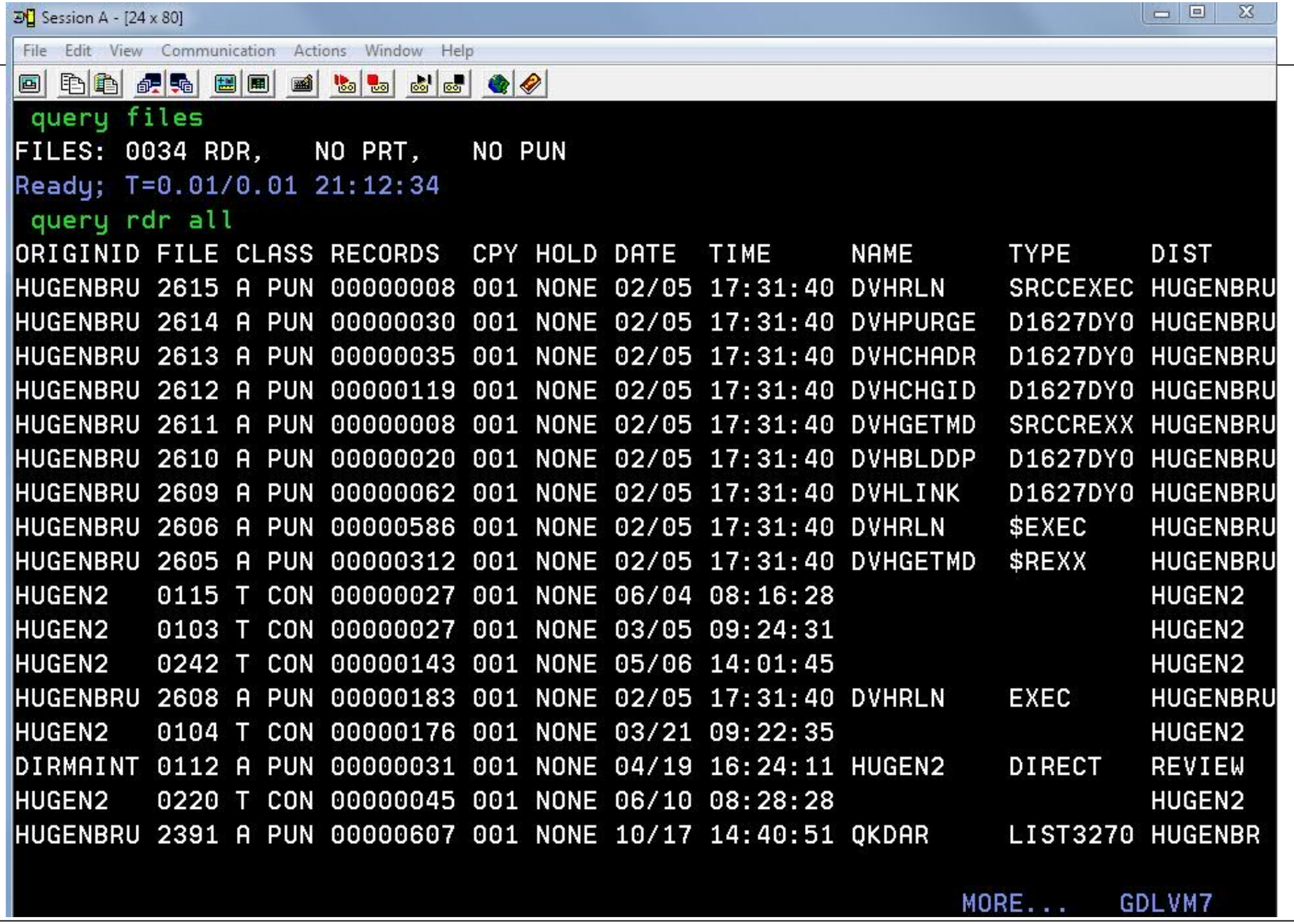


Defining a Virtual Machine
Sample User Directory Entry

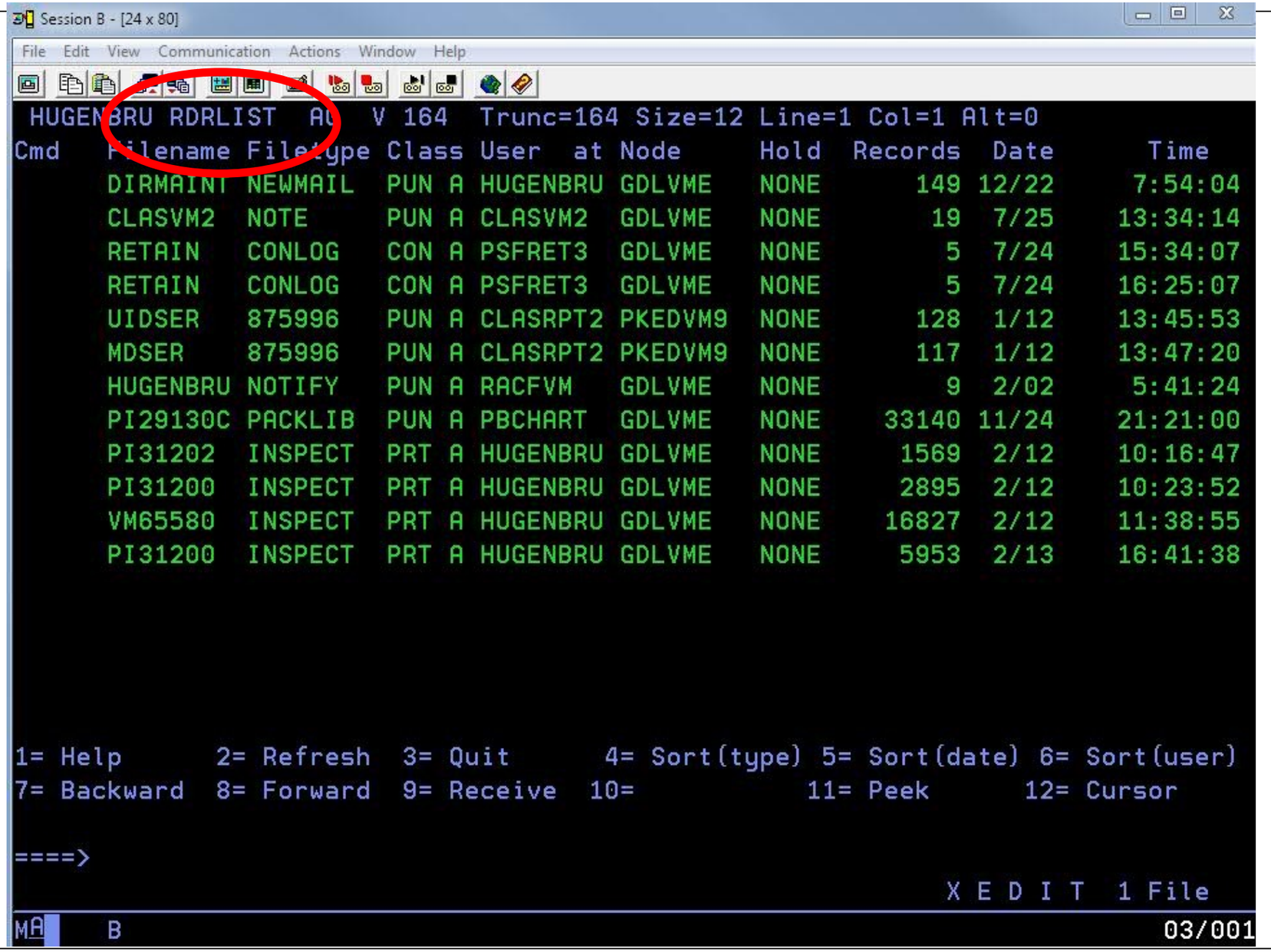
- Spool Devices:
  Virtualized Reader, Printer, and Punch



#IBMz #zVM



#IBMz #zVM

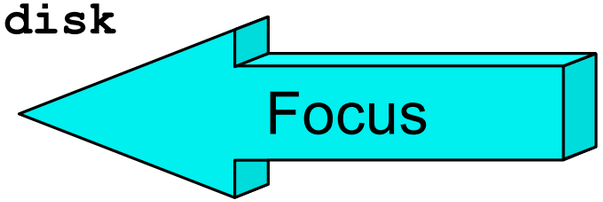


#IBMz #zVM

Defining a Virtual Machine

Sample User Directory Entry

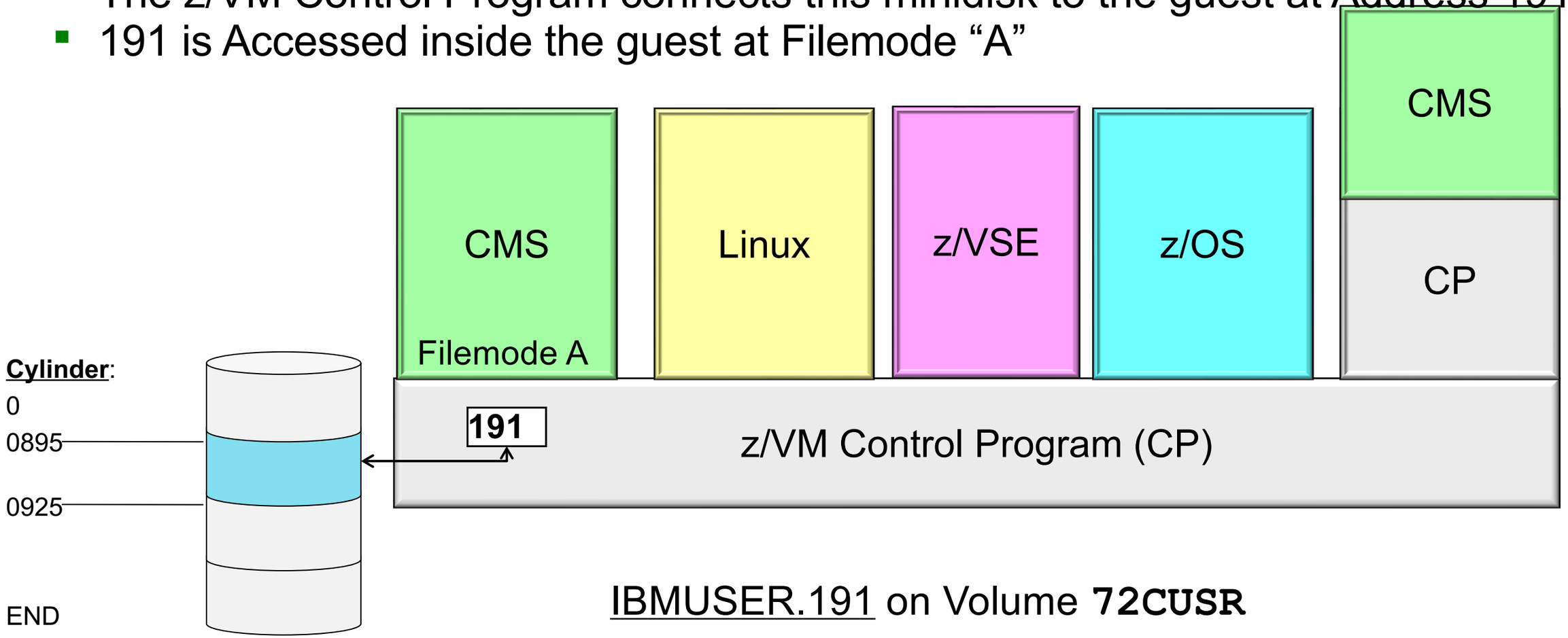
```
USER IBMUSER IBMUSER 32M 32M ABCDEG
 ACCOUNT SYSTEMS
 MACH ESA
 IPL CMS

 CONSOLE 009 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH A
 SPOOL 00E 1403 A

 LINK MAINT 0190 0190 RR * CMS system disk
 LINK MAINT 019E 019E RR * Product code disk
 LINK 7VMRAC20 29E 29E RR
 LINK 7VMRAC20 505 305 RR
 LINK 7VMRAC20 191 391 RR
 MDISK 1191 3390 2078 001 72CRES  MR READ    WRITE    MULTIPLE
 MDISK 191 3390 0895 030 72CUSR  MR READ    WRITE    MULTIPLE
```



Defining a Virtual Machine

Sample User Directory Entry

- Our CMS guest is running under z/VM
- The z/VM Control Program connects this minidisk to the guest at Address 191
- 191 is Accessed inside the guest at Filemode “A”



More About Minidisks

- A location on real DASD which has been allocated for storage of a user's files
- Three (3) types of minidisks:
  - Permanent
    - lasts across sessions (logons); defined in the User Directory
  - Temporary (T-disks)
    - Created inside a z/VM session
    - Destroyed at logoff!
    - use CP DEFINE command or ATTACH by operator
  - Virtual disks in storage (V-disks)
    - Temporary **simulation** of a minidisks in system storage
      - **not** allocated on real DASD (for example, on 3390 DASD)
    - Avoids I/O overhead
    - Good swap space for Linux guests

More About Minidisks – Commands

- CP DEFINE and FORMAT
  - Defines a virtual device or virtual disk in storage
    - `DEFINE T3390 as 815 cyl 10`
  - **Minidisks must be formatted before first time using!**
    - `FORMAT 815 D`

- CP LINK, CMS ACCESS
  - Link to other user's minidisks to share files
    - `LINK HUGENBRU 191 391 rr`
  - Once linked, a disk can be accessed
    - `ACCESS 391 C`

More About Minidisks – Commands

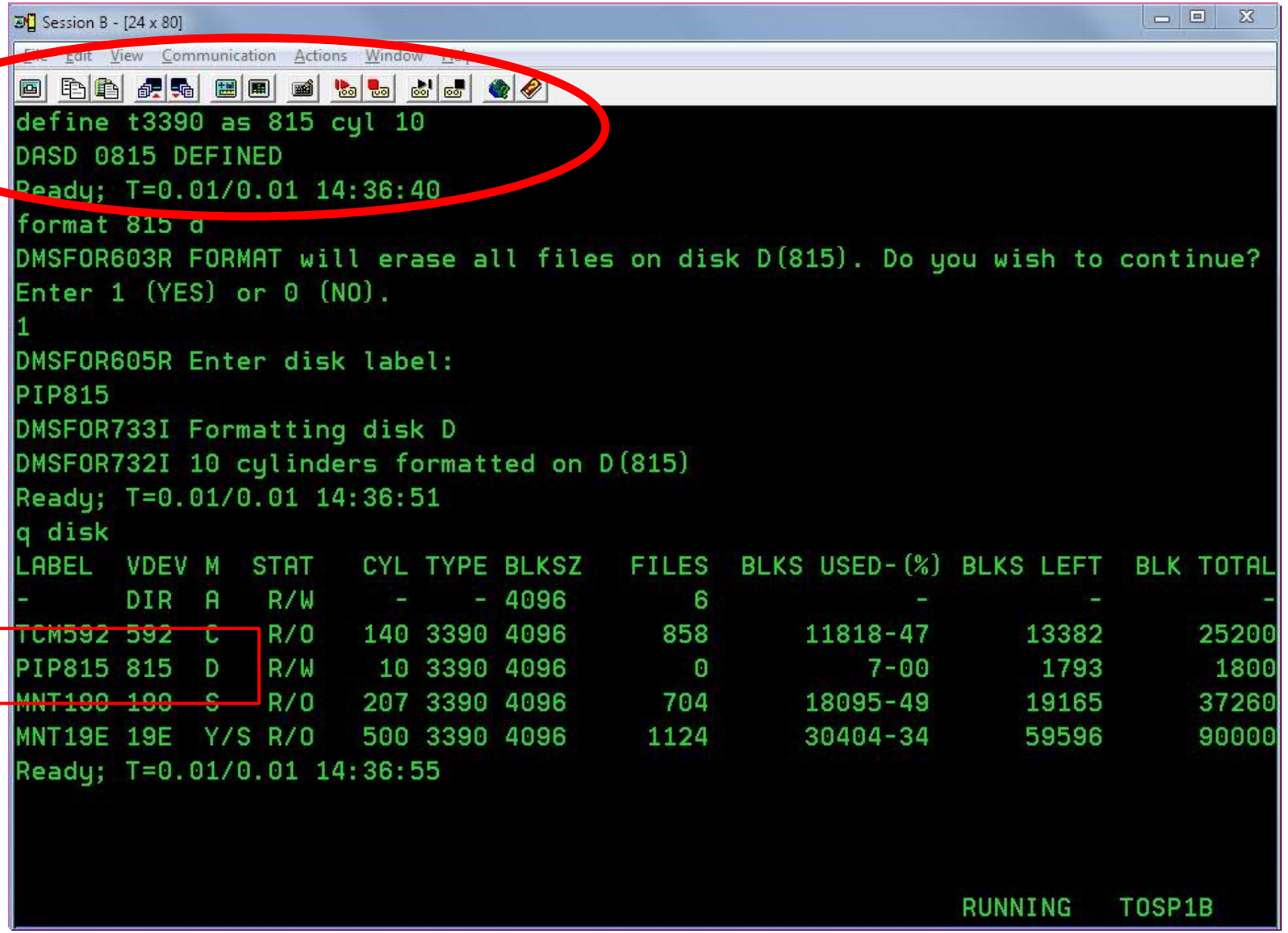
- RELEASE, CP DETACH
  - Release frees an accessed disk

```
-- Release C
```

  - Detach removes the device from your VM configuration

```
-- Detach 815 -or-   Release c (detach
```

- CP QUERY VIRTUAL DASD
  - Shows what your machine has linked; displays status
- Q ACCESSED, Q DISK, Q SEARCH
  - Shows various status information for accessed disks/directories
- LISTFILE, FILELIST
  - Lists the files on an accessed minidisk or directory



#IBMz #zVM

Developing Programs in a z/VM Environment

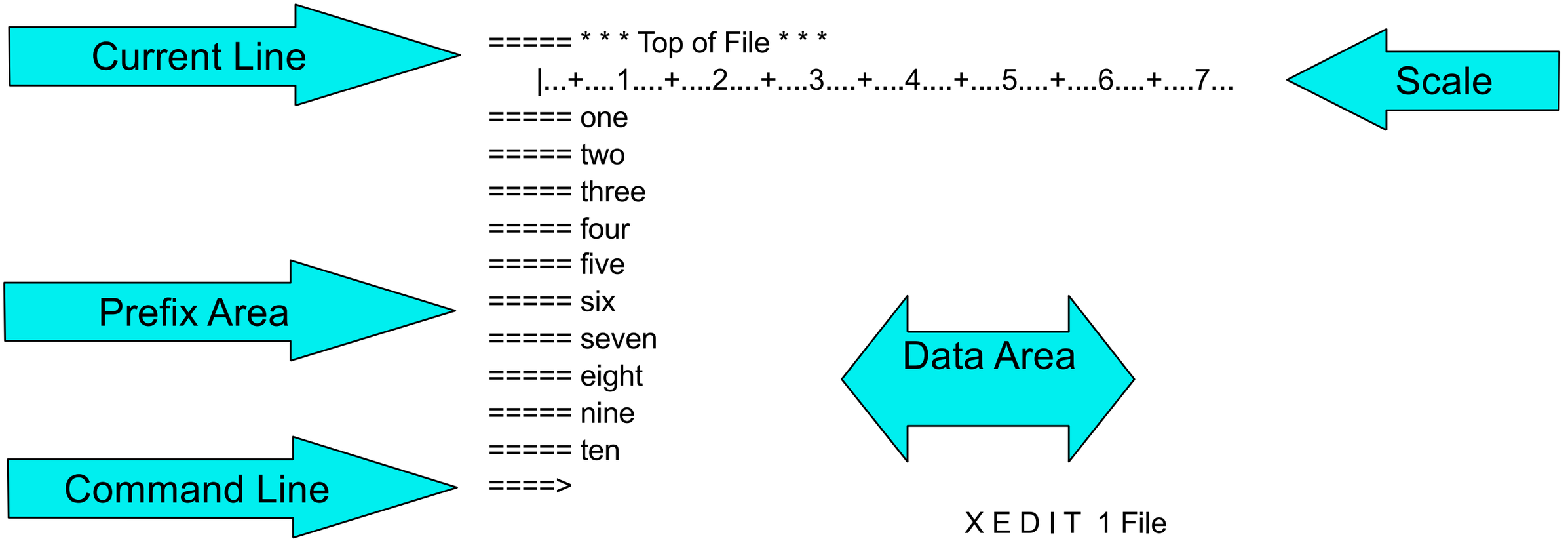
- Creating and Compiling
  - Filetype indicates name of programming language you are using
    - Assemble, Fortran, C, Cobol, PLI, Pascal, Rexx, etc.
  - Invoke the compiler by typing compiler name followed by File name of the program
  - LISTING and TEXT files are produced

    command: **`ASSEMBLE  ASM1`**

    result: **`ASM1 LISTING`**
    `ASM1 TEXT`
  - Use **XEDIT** to create the program like any other file

Developing Programs with XEDIT

- Command: `Xedit fn ft fm` TEST FILE A1 F 80 Trunc=80 Size=45 Line=0 Col=1 Alt=0
- Each line is a record
- Screen Layout:



Developing Programs with XEDIT

- Prefix Commands (subset)
  - `m, mm..mm` move, block move
  - `c, cc..cc` copy, block copy
  - `f` following
  - `p` preceding
  - `a` add
  - `si` sequential insert
  - `d, dd..dd` delete, block delete
  - `",""  ""` repeat previous command

Developing Programs with XEDIT

- Command-line commands
  - `screen format`
  - `change /xxx/yyy/ * *`
  - `/zzz/` (find zzz in text)
  - `all /zzz/` (find all zzz at once)

- Write your own XEDIT commands
  - Name: yourcmdn XEDIT
  - Write using REXX
  - Can use Pipelines

- Saving and Quitting your XEDIT Session
  - `QQuit` - leave without saving changes
  - `SAVE` - saves changes without exiting program
  - `FILE` - leave and save changes

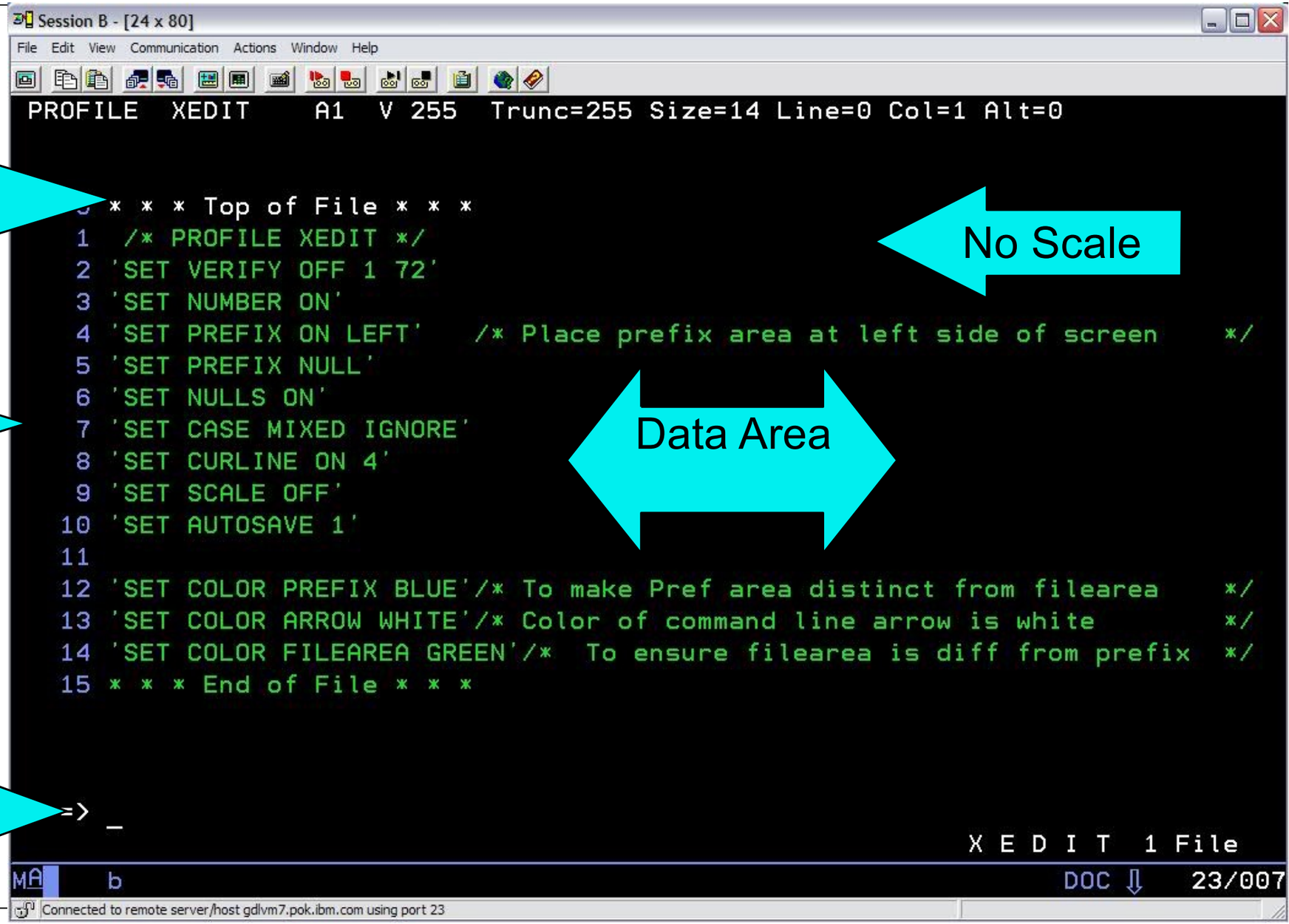
Developing Programs with XEDIT

- PROFILE XEDIT runs when XEDIT is invoked
- Sample: PROFILE XEDIT

```
/* PROFILE XEDIT */
'SET VERIFY OFF 1 72'
'SET NUMBER ON'
'SET PREFIX NULL'
'SET CASE MIXED IGNORE'
'SET CURLINE ON 4'
'SET SCALE OFF'
'SET AUTOSAVE 1'
```

- Note: Xedit is very tailorable !

Developing Programs with XEDIT



Developing Programs with XEDIT – Synonyms

- ISPF prefixes - PROFILE XEDIT

```
/* to mimic ISPF                  */
'SET PREFIX SYNONYM B P            '
'SET PREFIX SYNONYM A F            '
'SET PREFIX SYNONYM R "              '
'SET PREFIX SYNONYM RR ""          '
```

A Note on Programming in Rexx

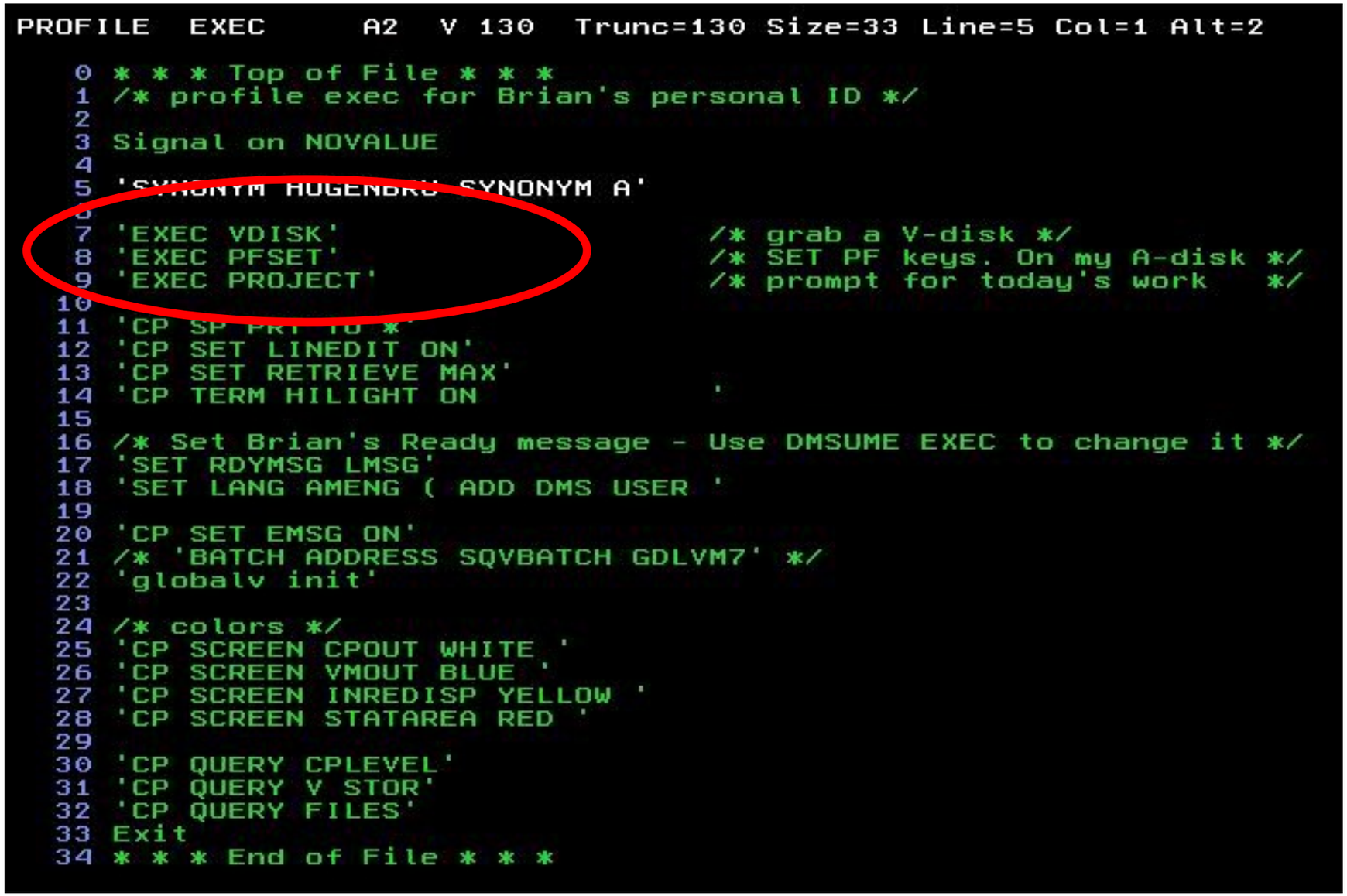
- Scripting language useful for writing productivity tools
  - Available on z/OS, Linux, Unix, Windows …
- Quick notes on REXX:
  - `/* always starts with a comment */`
  - Filetypes: EXEC (most common), EXEC2, REXX
  - Contains Variables and Stemmed Arrays
    - Stores Strings and Numbers as strings
  - Has 'flow control'
    - do and do…while
    - If…then, else, select
  - Allows Functions and Procedures
  - Allows for Pipelines
  - Issues CP/CMS commands (in quotes)

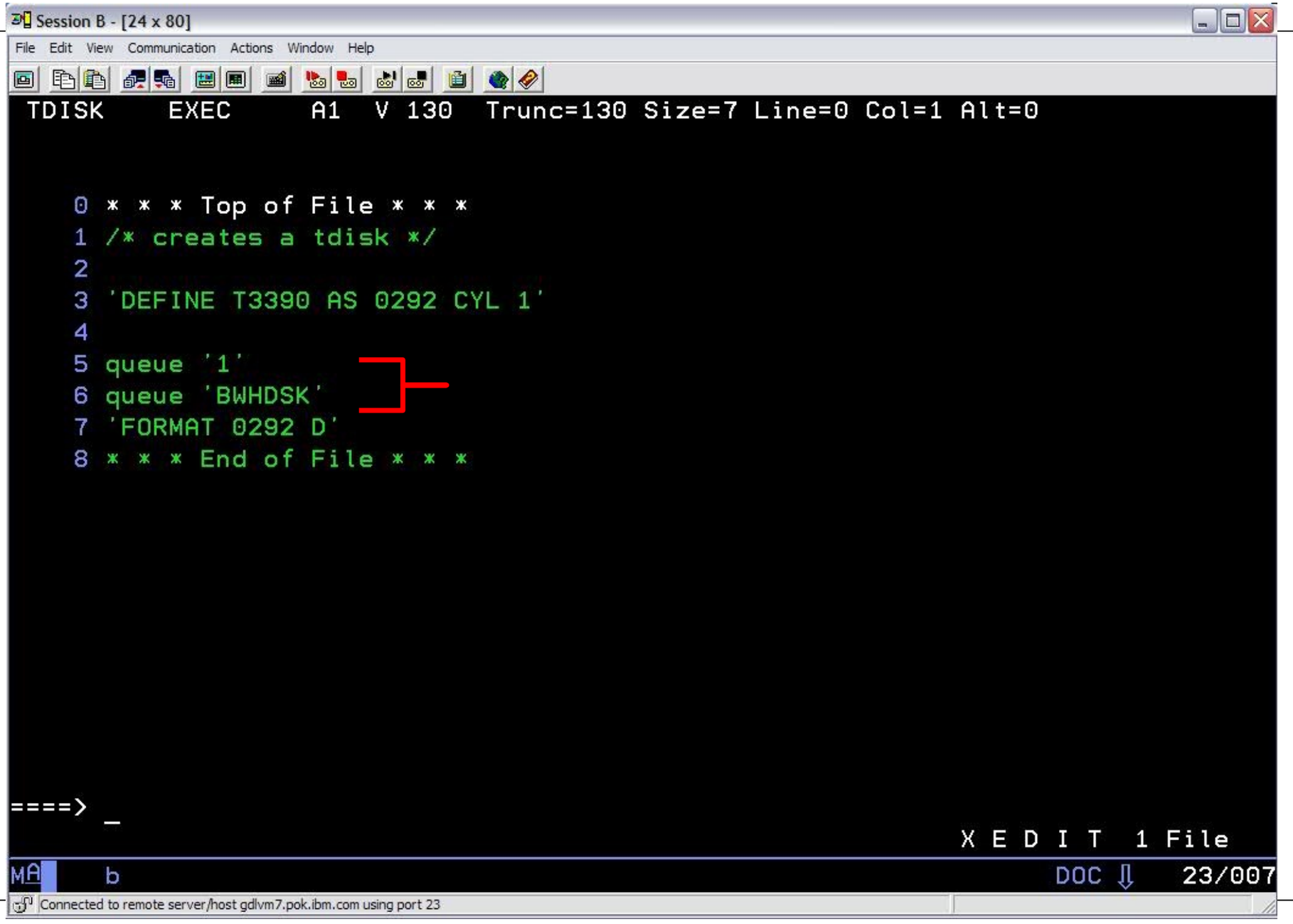
A Note on Programming in Rexx

- **`PROFILE EXEC`** runs when you sign on
- Contains all the tailoring and configuration desired for this virtual machine
- Can also issue commands at start-up

Try for yourselves! (5-minute exercise)

- `LISTFILE PROFILE EXEC *`
- `XEDIT PROFILE EXEC A`





Debugging A Virtual Machine

- Record your console
  - **`spool console to * start`** (to start recording)
  - **`...`** (do work here)
  - **`spool console stop close`** (punch console to reader)
  - **`query reader all`** (to get the spool id)
    - rdrlist
  - **`peek {spool id} (for *`** (to look at the console)
  - **`receive {spool id} fn ft fm`** (to save console in a file)

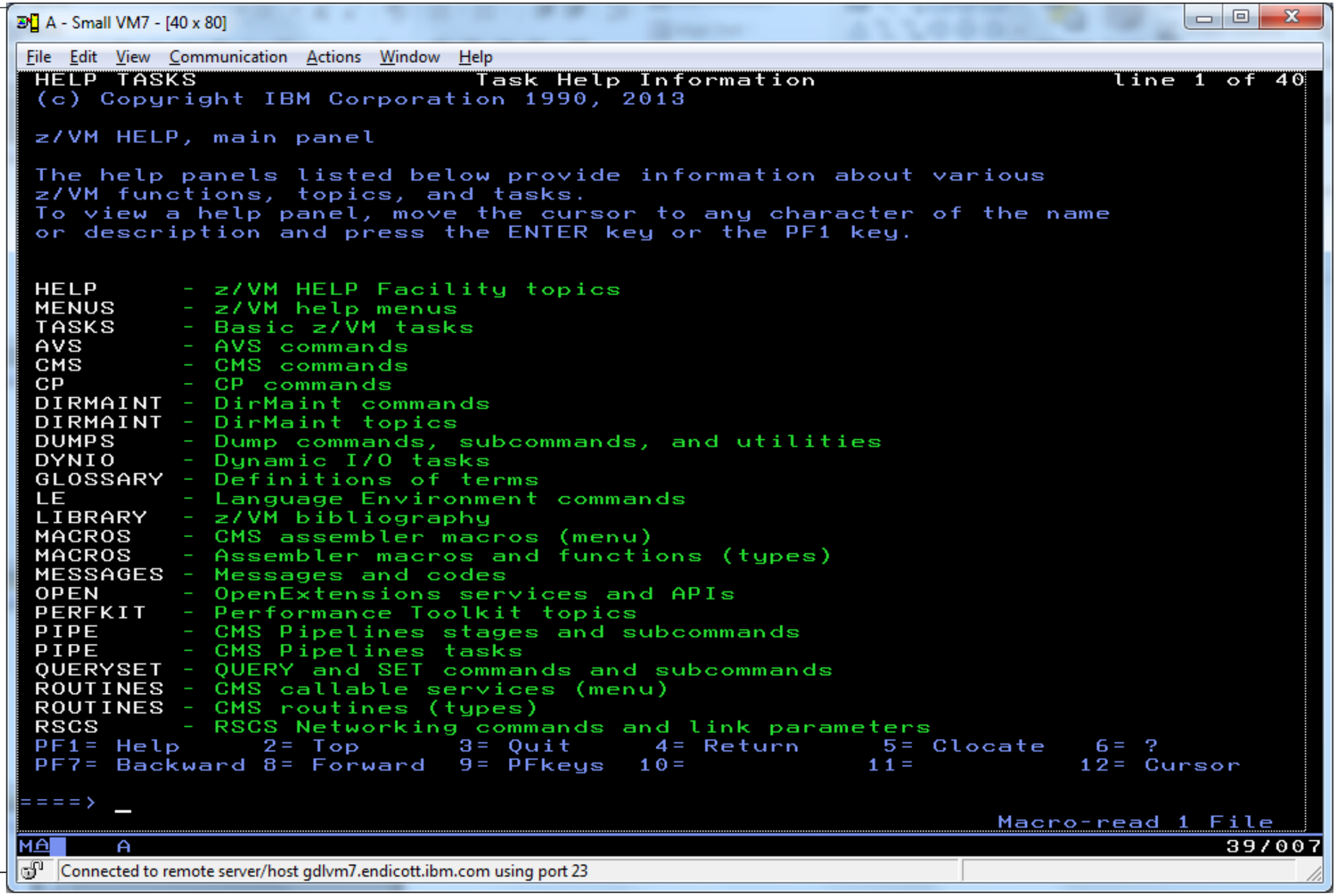
Debugging A Virtual Machine

- HELP Facility
  - HELP /* main menu */
  - HELP command /* for a specific command */
  - HELP msg DMSxxxE /* for a specific message */

- **`CP LINK MAINT 19D 19D RR`**
  - link for HELP disk

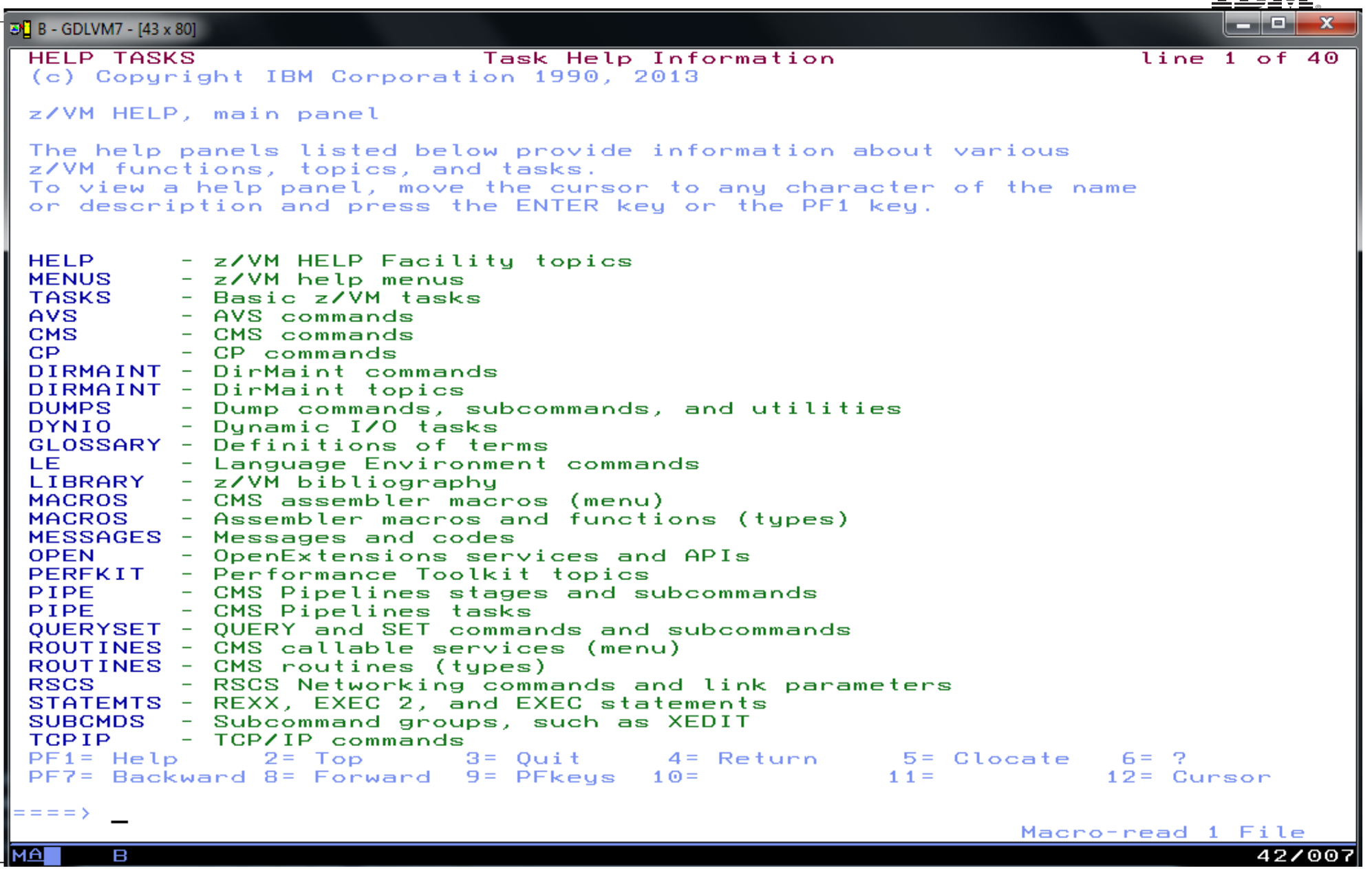
- *Most virtual machines that IBM ships by default already have this disk linked*

HELP

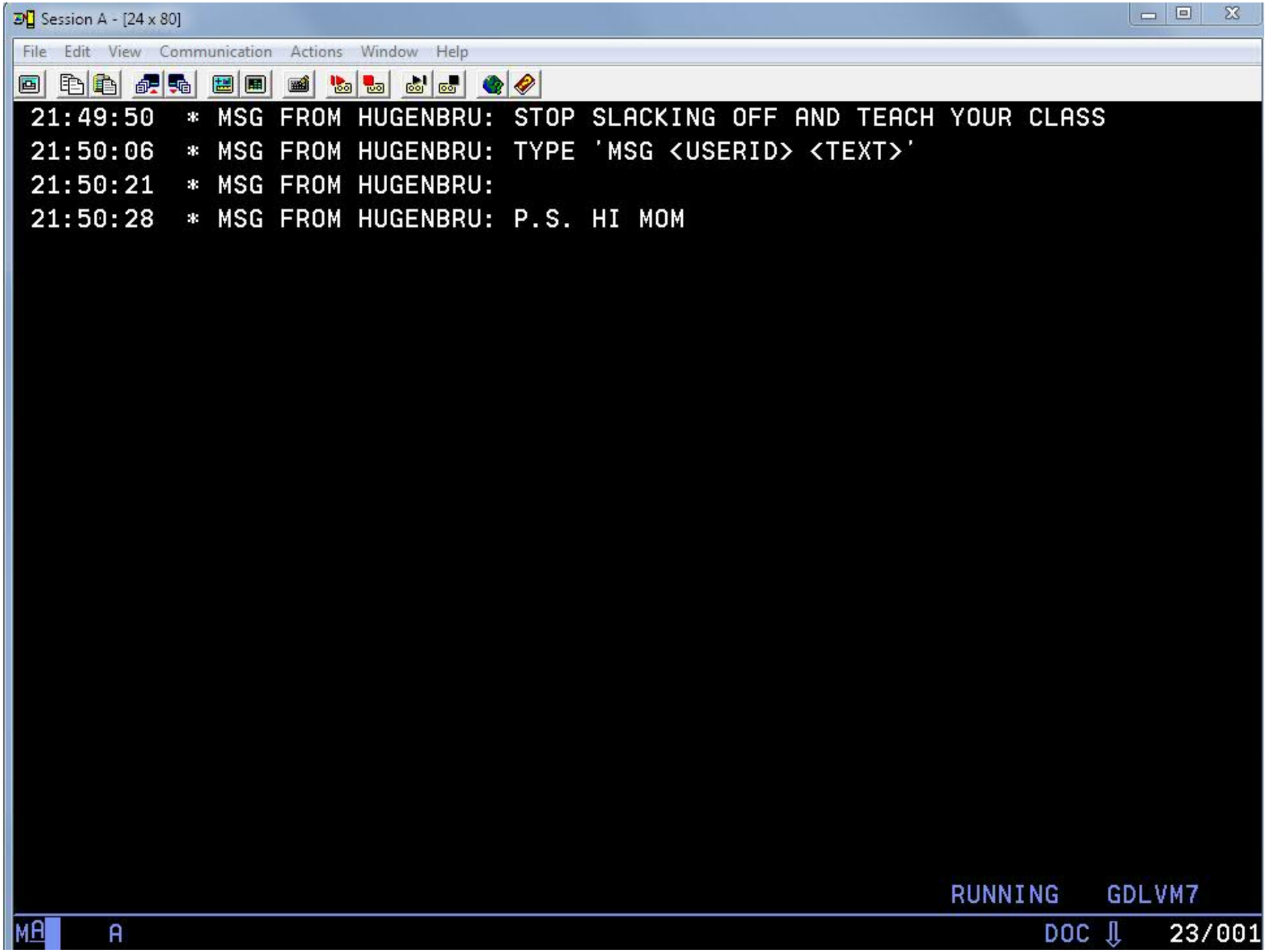


HELP

Note: the coloration of the help panels can be adapted for ease of reading.



#IBMz #zVM



Logging Off of a Virtual Machine

- **`CP LOGOFF`**
  - Terminates activity inside the virtual machine
  - Temporary and virtual disks are erased
  - User returns to logon screen

- **`CP DISCONNECT`**
  - Virtual machine and programs inside of it continue to run
  - User returns to logon screen

Try for yourselves!

- **SPOOL CONS TO * START**
- **QUERY ACCESSED**
- **QUERY LAN**
- **LISTFILE * EXEC ***
- **SPOOL CONS TO * STOP CLOSE**
- **QUERY RDR**
- **RDRLIST, *then hit PF11 on the last file to view***
- **HELP LINK**
  - **Q ACCESSED**
  - **LINK MAINT 19D 19D RR**
  - **ACC 19D <filemode>**
- **HELP LINK**
- **LOGOFF**

For more information …

z/VM Internet Library

- https://www.ibm.com/vm/library

- Includes:
  - z/VM Knowledge Center
    - Useful when unsure which book or for searching tasks
  - z/VM Information Center (being replaced by z/VM Knowledge Center)
  - z/VM PDF bookshelf
    - Useful for when you know the book you need and prefer PDFs
  - z/VM Program Directories
  - z/VM Education and Presentations
  - z/VM data areas and control blocks
  - z/VM monitor records
  - Select IBM Redbooks®
  - White papers and other documents

z/VM Library Overview

Last Updated: 17 December 2021

Update: Revised publications are now available for new or upcoming z/VM 7.2 New Function APARs released between September and December 2021. An updated collection kit containing all the updates is also available.

You can now get all IBM z/VM PDF files directly from the z/VM Library web site. All documentation pertaining to a specific supported version and release will be available regardless of the New Function APARs you have installed on your z/VM system. PDF files are available for z/VM 6.x and z/VM 7.1 releases as well as z/VM related publications.

The z/VM Library also offers complete PDF collections of the documentation on a near quarterly basis. These collections are indexed, meaning you can search for any terms and it will return all matches from any document within the collection. No more hunting for a specific manual to find exactly what you need.

z/VM product information

- → z/VM 7.2 PDFs
- → z/VM 7.1 PDFs
- → z/VM 6.x PDFs
- → z/VM Related PDFs

z/VM CMS Primer

- Primer walks through getting started with CMS
- Key content:
  - –Logging on and off of z/VM
  - –CMS minidisk file system and SFS (Shared File System) background
  - –Editing files using XEDIT
  - –Managing files and several productivity aids included in CMS for this
    - Like other platforms, there is often more than one way to do things.
- Content you can probably skip when you read for first time:
  - –Printing from CMS
  - –NAMES files

https://www.vm.ibm.com/library/720pdfs/72626501.pdf [PDF]

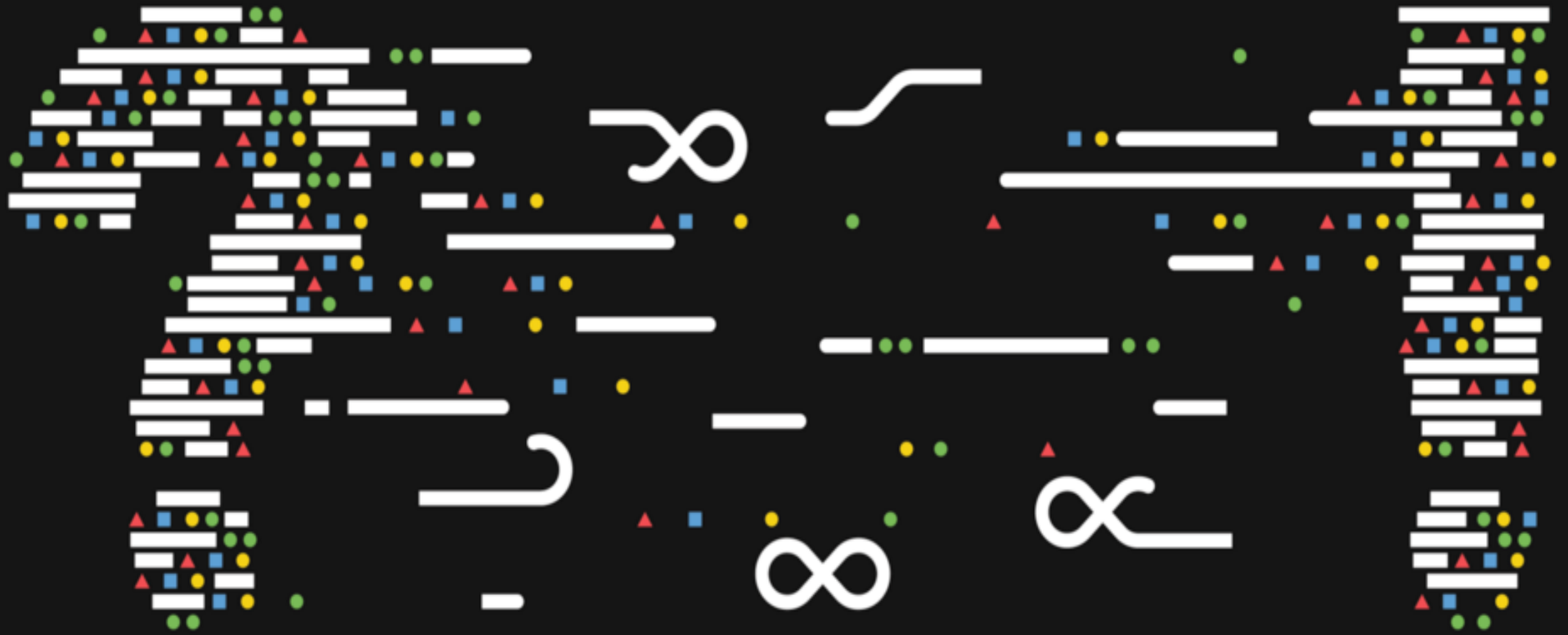
For More Information …

- **Web sites:**
  - https://www.vm.ibm.com/ -- zVM on the Web
  - https://www.vm.ibm.com/library -- the online zVM Library
  - https://www.vm.ibm.com/library/presentations/ -- links to presentations like this one!
  - https://www.youtube.com/playlist?list=PL_4RxtD-BL5uGoq629H8IDxzfuvAN3IpF
    - "z/VM How-To Guides" YouTube Channel

- **Via mailing lists:**
  - IBMTCP-L@VM.MARIST.EDU
  - IBMVM@LISTSERV.UARK.EDU
  - LINUX-390@VM.MARIST.ED

Contact Information:

Brian W. Hugenbruch, CISSP
IBM zSystems Security for Virtualization and Cloud
bwhugen at us dot ibm dot com
@Bwhugen



Backup Slides

Debugging A Virtual Machine

- Tracing
  - trace i r12345.10
    - trace instructions @ location 12345 for x'10' bytes
  - display g (display general registers)
  - d t12345.20 (display translated storage for x20)
  - b (begin execution)
  - trace end (end tracing)

- Dumps
  - VMDUMP
  - VM Dump Tool

Try for yourselves!

- **LOGON <userid>** /* Remember, CTRL is your Enter key */
- **SPOOL CONS TO * START**
- **QUERY PRIVCLASS** /* Your security context ... */
- **QUERY COMMANDS**
- **QUERY DISK** /* Your resources ... */
- **QUERY ACCESSED**
- **QUERY LAN**
- **SPOOL CONS TO * STOP CLOSE**
- **RDRLIST, *then hit PF11 on the last file to view … look familiar?***
- **PURGE RDR ALL**
- **QUERY RDR**
- **HELP LINK**
- **LINK MAINT 19D 19D RR**
- **HELP LINK**
- **LOGOFF**

FAQ: “What are those numbers on my filemode?”

- You may see a second character (0-6) appear in the *filemode* field when doing a FILELIST on your minidisk or SFS directory. It’s okay, it’s meant to be there.
  - The number is a functional marker. The default is ‘1’. CMS system disks are a special case and should be ‘2’.
  - Function sometimes differs depending on if it’s a minidisk or an SFS directory:

	Minidisk	SFS
0	File is private, unless a user has R/W access to the disk	No meaning
1	Default: normal read and/or write access	
2	Same Function as 1; conventionally used for files on shared disks.	Same Function as 1; no conventional purpose.
3	File is erased after it is read (be careful with this one)	
4	OS simulated data set format	OS simulated data set format
5	Same Function as 1; conventionally used for filegroups.	Same Function as 1; no conventional purpose.
6	Indicates “Update in place” is in effect.	Same Function as 1; no conventional purpose. (Update-In-Place handled via extended file attributes)

Brian's Cheat Sheet v1.2: "Where is it?"
(z/VM 7.2 edition)

File	Where It Is	What It Is
`LOGO.CONFIG`	`PMAINT CF0`	z/VM Logo Screen
`SYSTEM.CONFIG`	`PMAINT CF0`	z/VM System Configuration File
`CPLOAD.MODULE`	`MAINT CF1` `MAINT CF3 (backup)`	z/VM Control Program (the "kernel")
`USER.DIRECT`	`PMAINT 2CC`	All virtual machine definitions (Note: resides in different place if you're using a Directory Manager)
Help Files	`MAINT 19D`	Access to HELP command
Sample Utilities	`MAINT720 2C2`	Access to z/VM utilities