IBM GLOBAL SERVICES

IBM®

Session B33

z/VM's Control Program (CP)
Part 2 - Under the Covers

John Franciscovich
francisj@us.ibm.com

zSeries® EXPO
FEATURING Z/OS, Z/VM, Z/VSE AND LINUX ON ZSERIES

September 19 - 23, 2005

San Francisco, CA

RETURN TO INDEX

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

- CICS*
- DB2*
- DB2 Connect
- DB2 Universal Database
- DFSMS/MVS*
- DFSMS/VM*
- e business(logo)*
- eServer
- Enterprise Storage Server*
- ESCON*
- FICON
- GDDM*
- HiperSockets
- IBM*
- IBM(logo)*
- Language Environment*
- MQSeries*
- Multiprise*
- MVS
- NetRexx
- OpenEdition*
- OpenExtensions
- OS/390*
- Parallel Sysplex*
- PR/SM
- QMF
- RACF*
- RAMAC*
- RISC
- S/370
- S/390*
- S/390 Parallel Enterprise Server
- VisualAge*
- VisualGen*
- VM/ESA*
- VTAM*
- VSE/ESA
- WebSphere*
- z/Architecture
- z/OS*
- zSeries*
- z/VM*

* Registered trademarks of the IBM Corporation

The following are trademarks or registered trademarks of other companies.

Lotus, Notes, and Domino are trademarks or registered trademarks of Lotus Development Corporation.
Tivoli is a trademark of Tivoli Systems Inc.
Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States, other countries, or both.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation in the United States, other countries, or both.
Penguin (Tux) compliments of Larry Ewing

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

IBM considers a product “Year 2000 ready” if the product, when used in accordance with its associated documentation, is capable of correctly processing, providing and/or receiving date data within and between the 20th and 21st centuries, provided that all products (for example, hardware, software and firmware) used with the product properly exchange accurate date data with it. Any statements concerning the Year 2000 readiness of any IBM products contained in this presentation are Year 2000 Readiness Disclosures, subject to the Year 2000 Information and Readiness Disclosure Act of 1998.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Disclaimer

The information contained in this document is not intended to be an assertion of future action by IBM. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adopt these techniques to their own environment do so at their own risk.

In this presentation, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used instead.

Any performance data contained in this presentation was determined in a controlled environment and, therefore, the results which may be obtained in other operating environments may vary significantly. Users of this presentation should verify the applicable data for their specific environment.

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

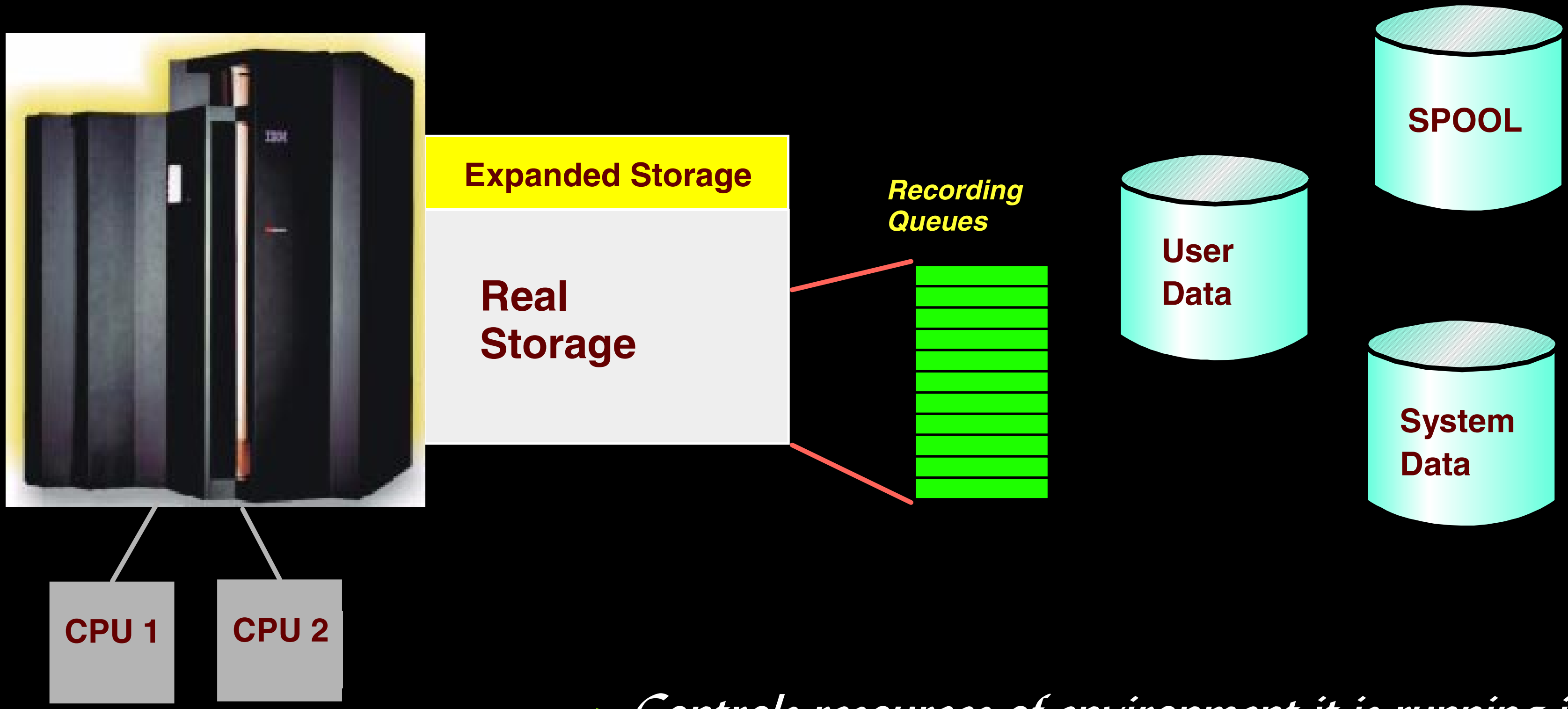
Any feedback that you give IBM regarding this presentation will be treated as non-confidential information. IBM reserves the right to use this information in any form.

Topics

- Overview (review) of z /VM's CP
- CP Startup Process
- Storage (Memory) and SPOOL Management
- Running Virtual Machines
- Shutting Down CP
- Collecting Diagnostic Data

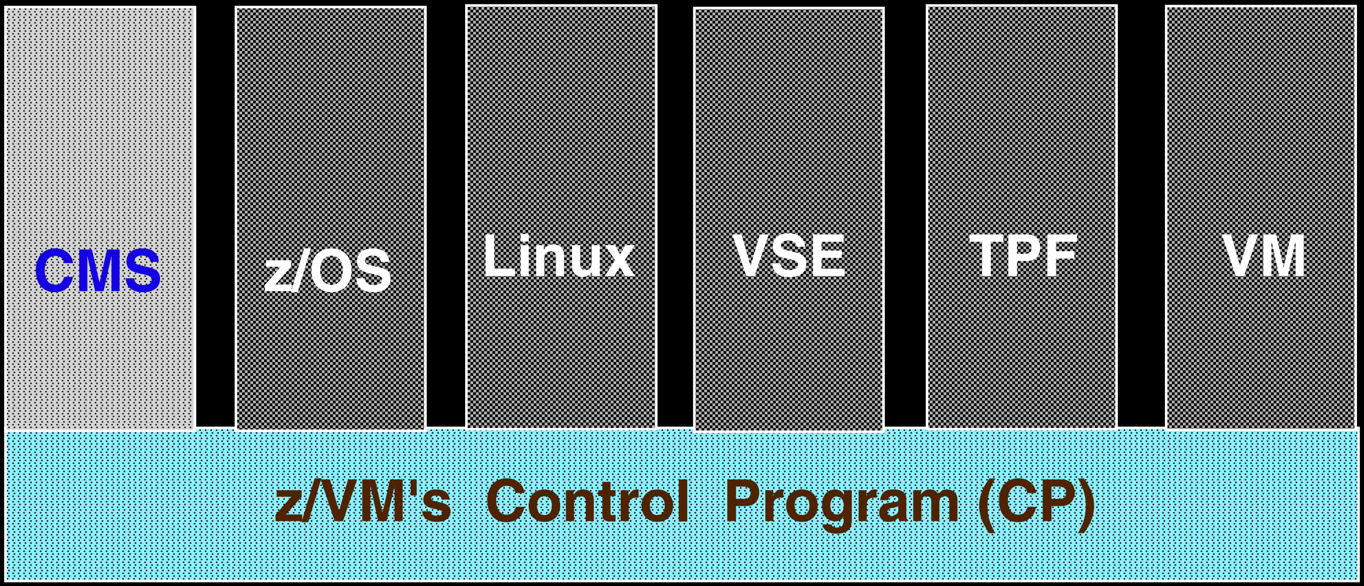
Overview

CP - z/VM's System Control Program



- *Controls resources of environment it is running in*
  - ➡ Native
  - ➡ LPAR
  - ➡ Virtual machine
- *Manages storage (memory) and devices*
- *Records usage and system event data*
- *Provides error recovery facilities*

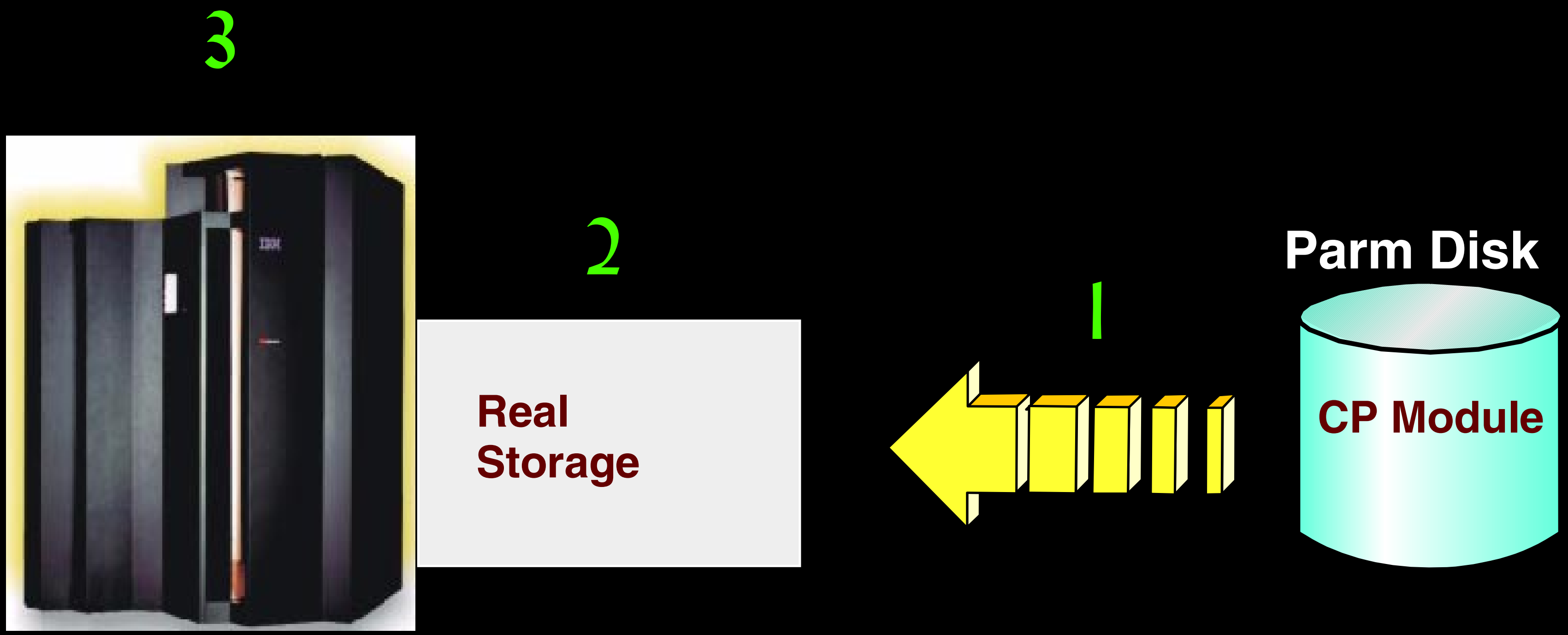
CP - z/VM's System Control Program...



- *Manages virtual machines*
  - ESA/390 and z/Architecture
  - Guest operating systems
  - Interactive users
    - *CMS is a special single user operating system that is part of z/VM*
- *Shares real resources among virtual machines*
- *Supports connectivity among virtual machines*
  - Virtual networking
  - Data sharing and exchanging information

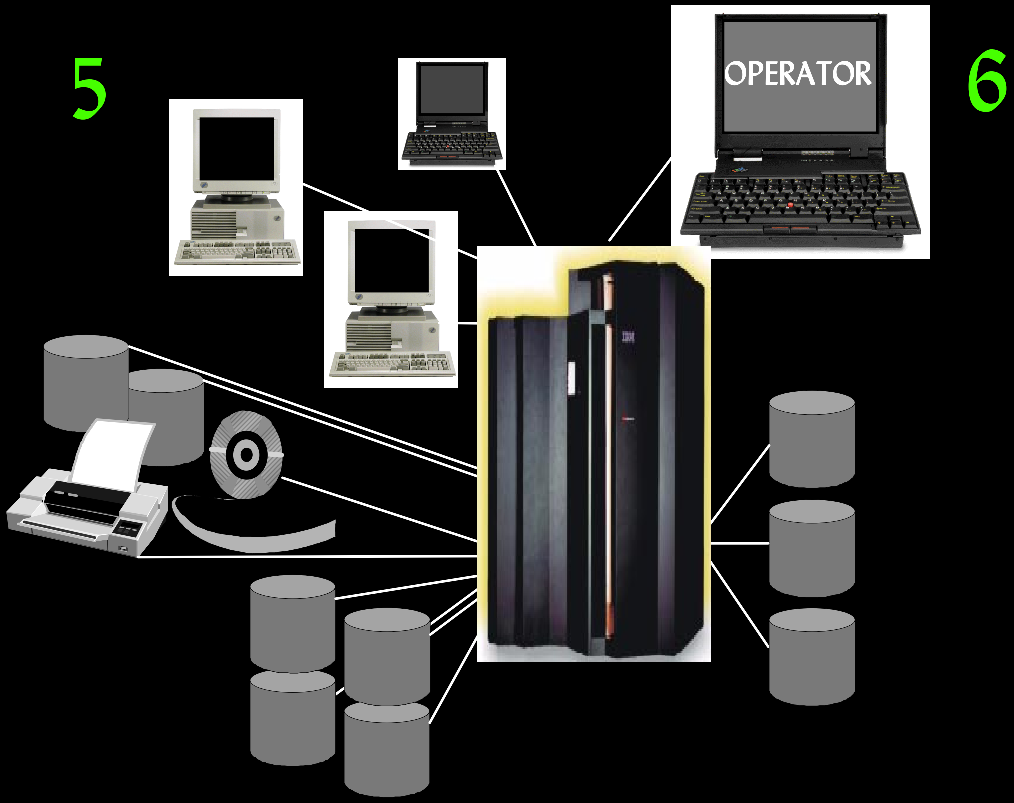
CP's Startup Process

Initializing CP



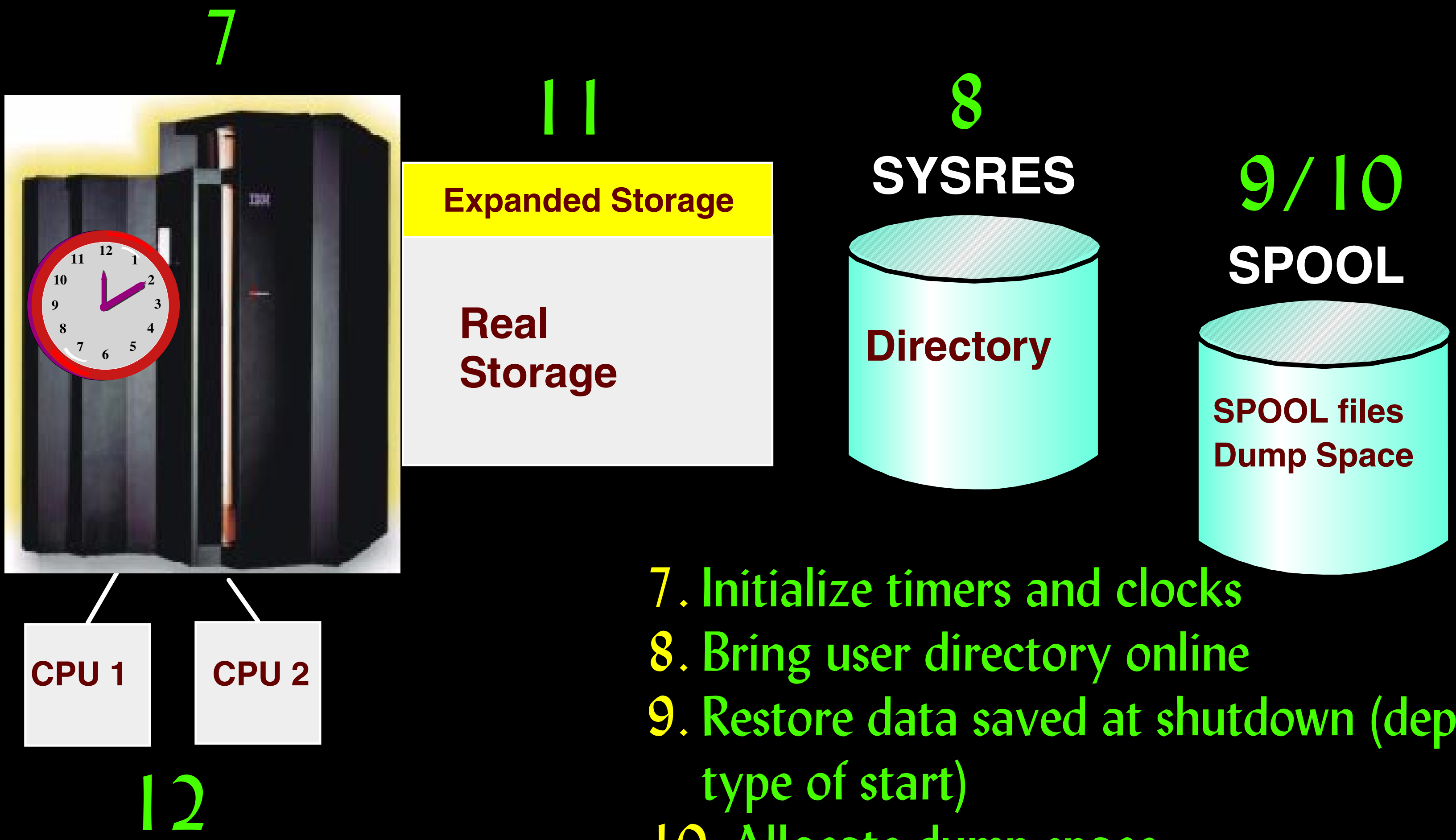
1. Stand Alone Program Loader (SAPL) loads CP Module into storage
2. CP real and virtual storage are initialized
3. Environment configuration information is obtained and saved

Initializing CP...



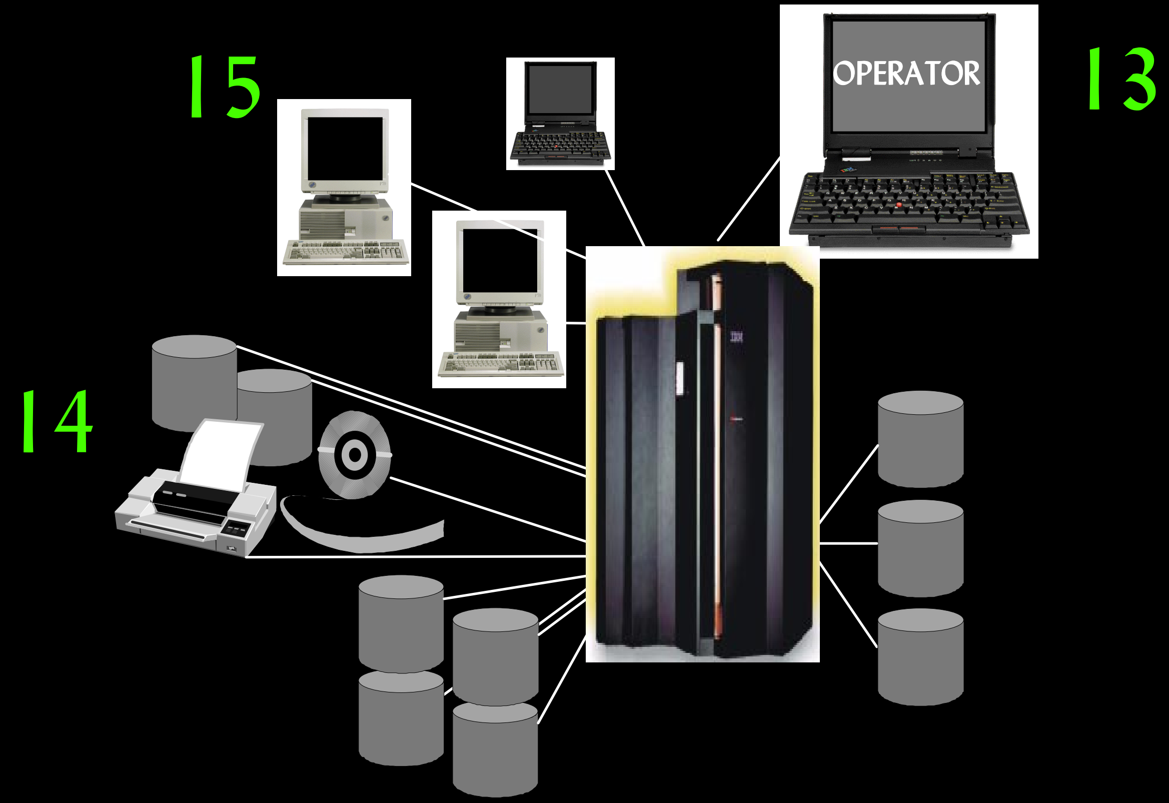
5. Initialize all available and system generated I/O devices
6. Locate OPERATOR's console

Initializing CP...



7. Initialize timers and clocks
8. Bring user directory online
9. Restore data saved at shutdown (depending on type of start)
10. Allocate dump space
11. Initialize expanded storage
12. Bring additional CPUs online

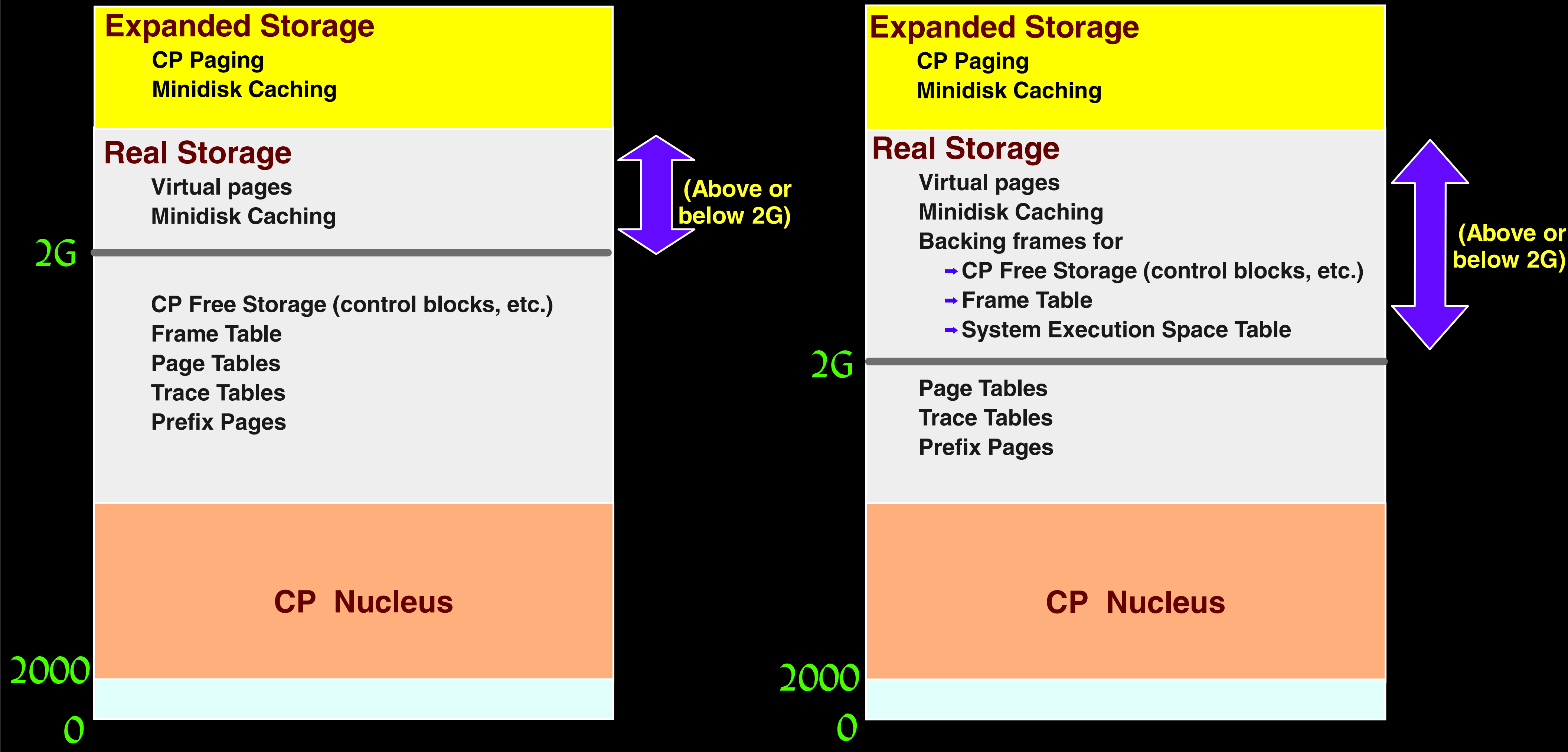
Initializing CP...



13. Log on the system operator
14. Start real spooling devices
15. Enable all terminal devices

Storage (Memory) and SPOOL Management

CP's Storage (Memory) Usage



z/VM 5.1.0

Expanded Storage
- CP Paging
- Minidisk Caching

Real Storage
- Virtual pages
- Minidisk Caching

(Above or below 2G)

2G

- CP Free Storage (control blocks, etc.)
- Frame Table
- Page Tables
- Trace Tables
- Prefix Pages

CP Nucleus

2000

0

z/VM 5.2.0

Expanded Storage
- CP Paging
- Minidisk Caching

Real Storage
- Virtual pages
- Minidisk Caching
- Backing frames for
  - CP Free Storage (control blocks, etc.)
  - Frame Table
  - System Execution Space Table

(Above or below 2G)

2G

- Page Tables
- Trace Tables
- Prefix Pages

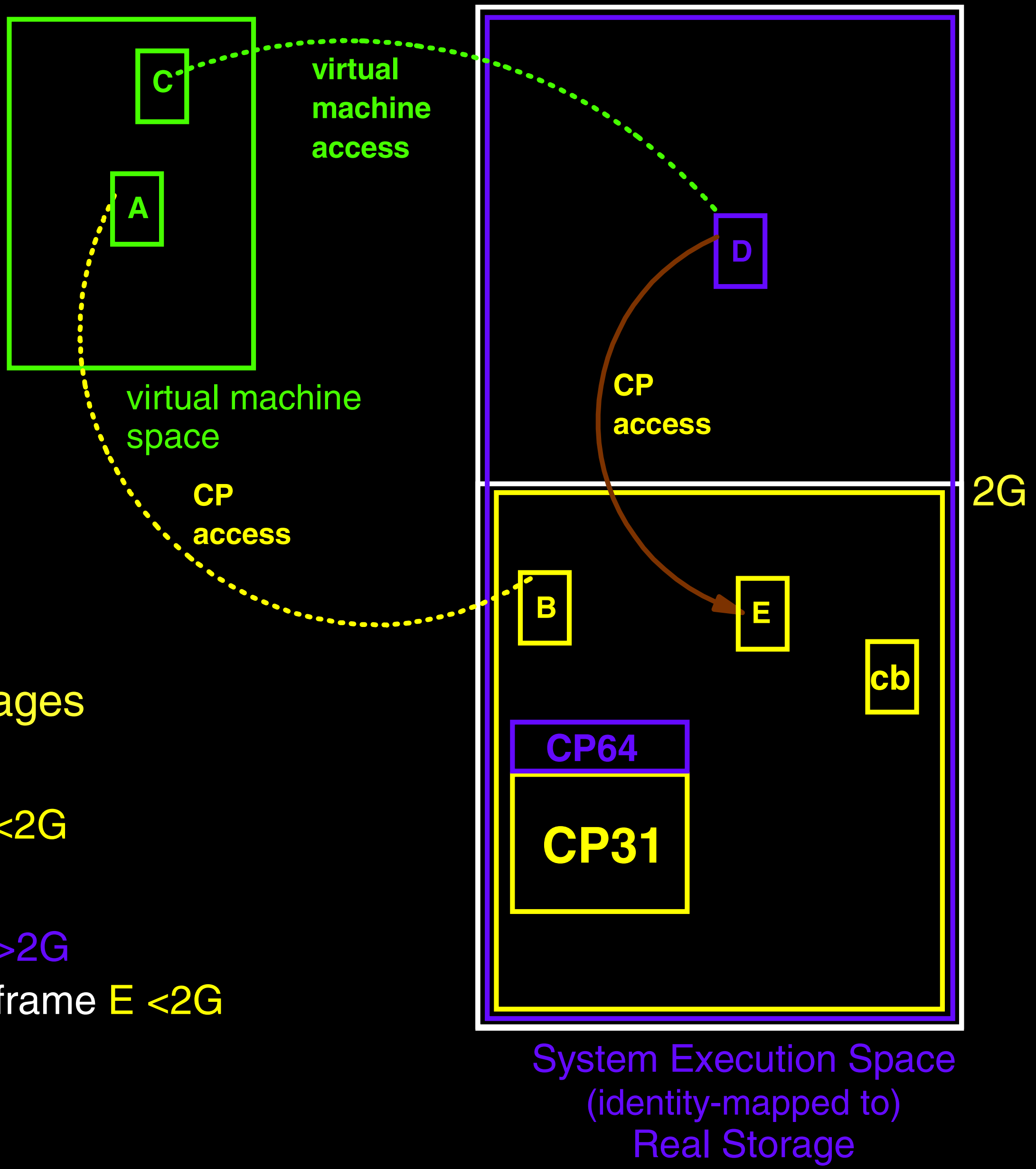
CP Nucleus

2000

0

CP Storage Usage (z/VM 3.1.0 - 5.1.0)

(64-bit CP)



CP references to virtual machine pages

- Page A
  - resides in host page frame B <2G
- Page C
  - resides in host page frame D >2G
  - must be **moved** to host page frame E <2G

CP Storage Usage (z/VM 3.1.0 - 5.1.0)...

64-Bit CP build

- Limited exploitation of storage >2G
- Mostly 31-bit addressing mode ("CP31")
- Small amount in 64-bit addressing mode ("CP64")

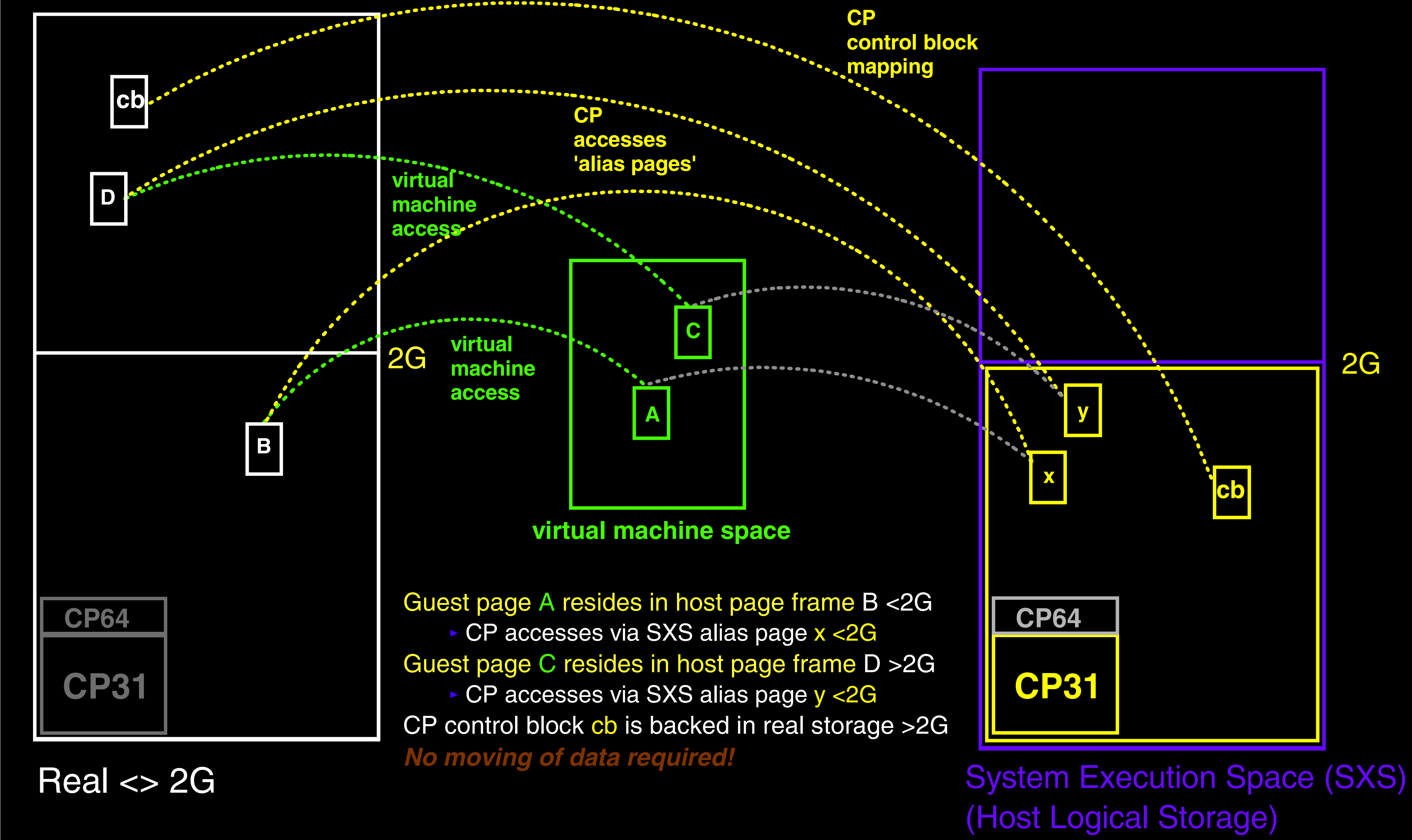
Virtual machine pages can reside >2G

- Must be moved <2G to be referenced by CP

All CP owned structures must be <2G

- Free Storage
- Control Blocks

CP Storage Usage - 64-Bit Exploitation (z/VM 5.2.0)



Guest page A resides in host page frame B <2G

- CP accesses via SXS alias page x <2G

Guest page C resides in host page frame D >2G

- CP accesses via SXS alias page y <2G

CP control block cb is backed in real storage >2G

No moving of data required!

CP Storage Usage - 64-Bit Exploitation (z/VM 5.2.0)...

64-Bit CP

- Mostly 31-bit addressing mode ("CP31")
  - *References storage <2G in the System Execution Space (Host Logical)*
  - *Can implicitly reference storage >2G*
- Parts of CP execute in 64-bit addressing mode ("CP64")
  - *Explicitly reference real storage >2G*

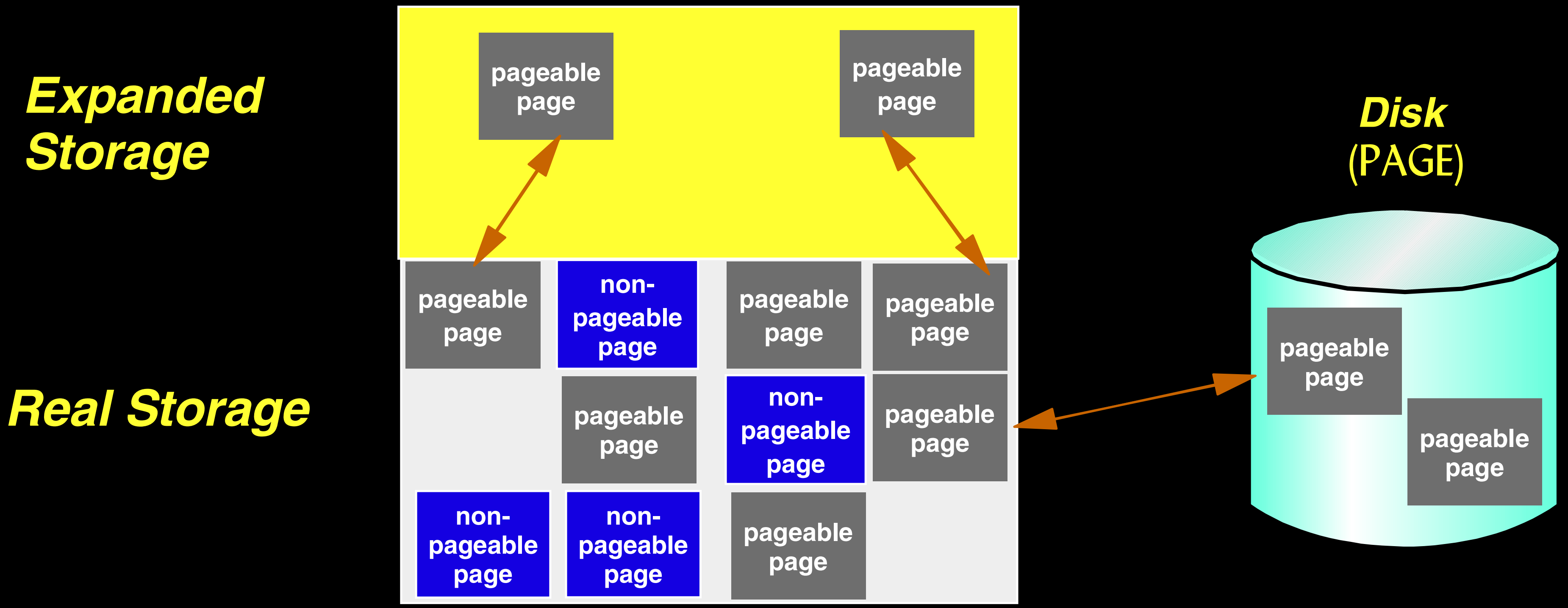
Virtual machine pages can reside >2G

- Mapped to "alias" page <2G to be referenced by 31-bit CP
  - *31-bit Host Logical address*

CP owned structures can reside in real storage >2G

- Free Storage
- Control Blocks

Managing Real Storage Among Virtual Machines



CP optimizes use of real storage for virtual machines

- Virtual machine storage is pageable
  - *Demand paged* - only paged out when necessary
- Paged to
  - *Expanded storage*
  - *Disk (CP-Owned PAGE area)*

Managing Real Storage Among Virtual Machines…

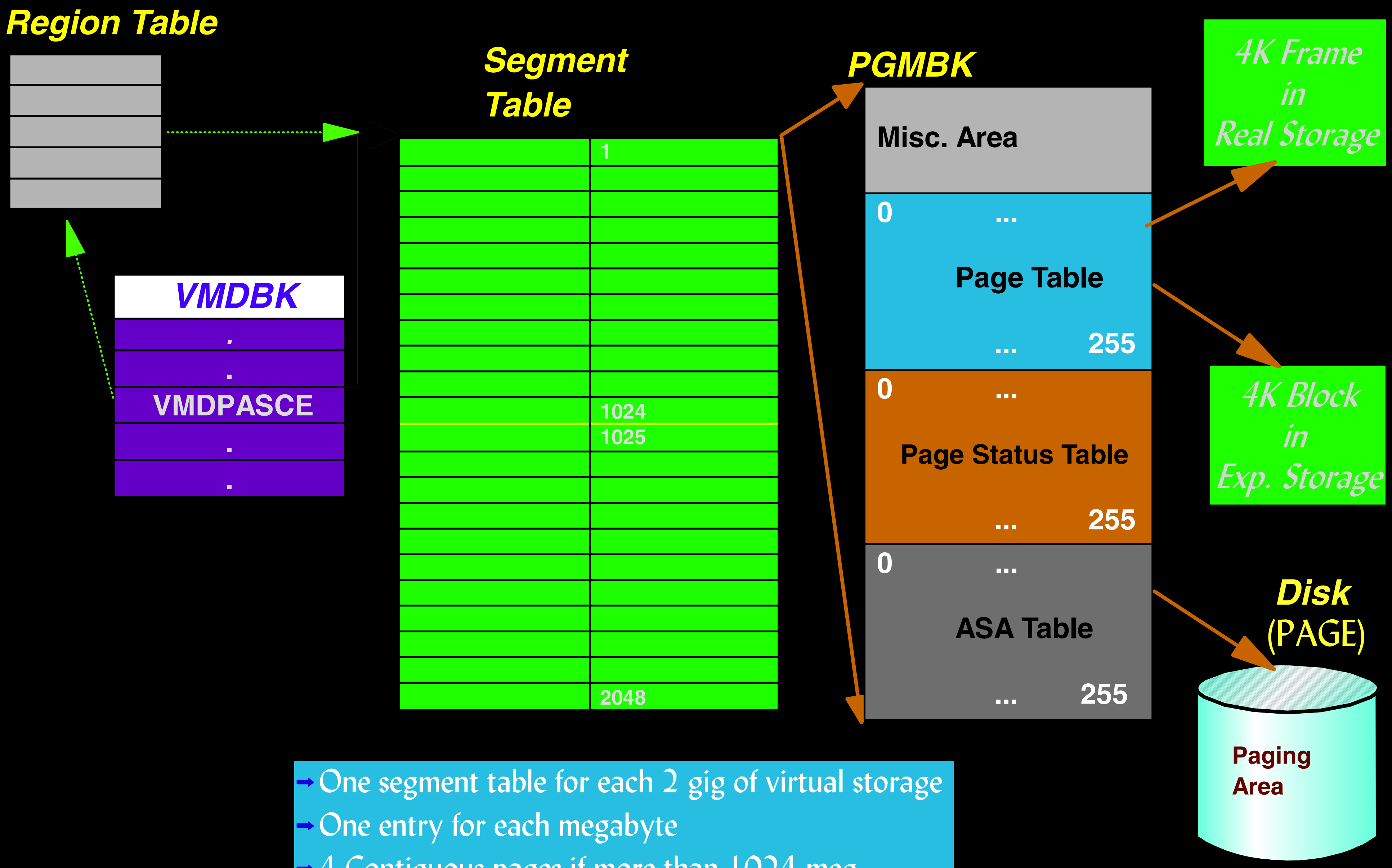
Non-Pageable Pages - Examples

- CP nucleus
- Prefix pages for alternate processors
- Frame and page tables
- CP free storage

Pageable Pages - Examples

- Virtual machine storage
- Spool buffers
- Virtual disks
- Trace tables

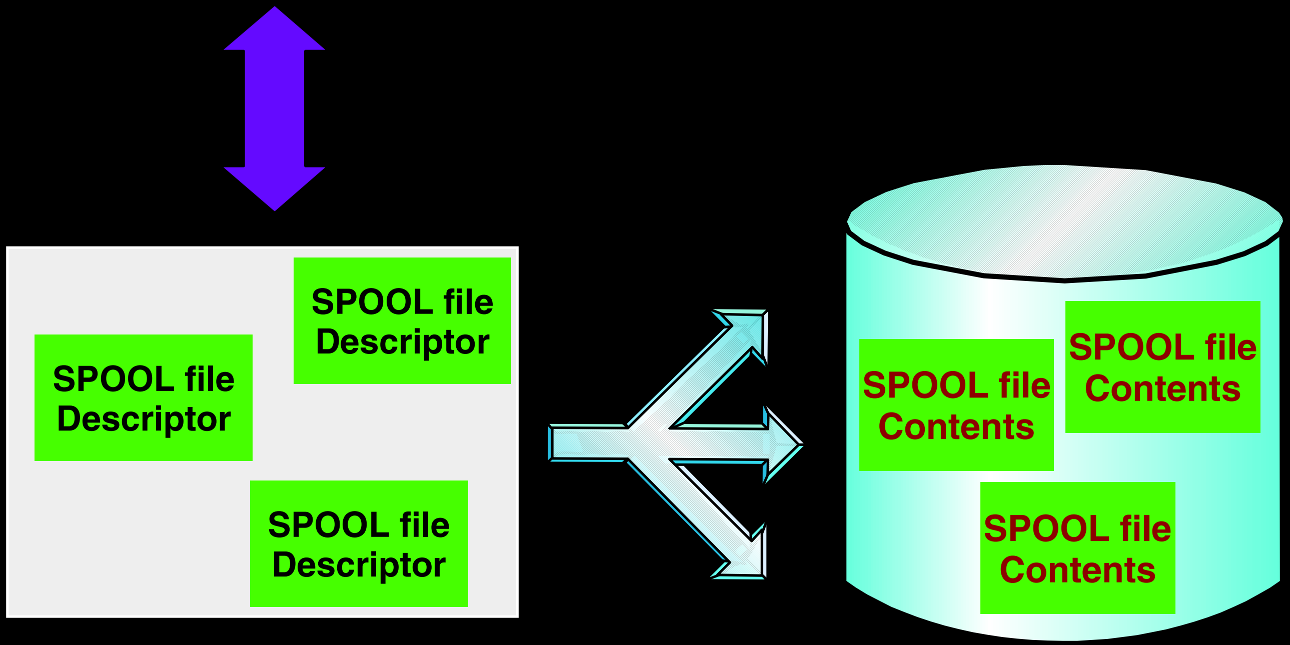
A Virtual Machine's Storage



- One segment table for each 2 gig of virtual storage
- One entry for each megabyte
- 4 Contiguous pages if more than 1024 meg

CP SPOOLing

```
q rdr all
ORIGINID  FILE CLASS RECORDS  CPY HOLD DATE  TIME      NAME     TYPE     DIST
OPERATOR  0039 A PUN 00000089 001 NONE 09/02 15:50:06  PROFILE  EXEC     35H:0253
OPERATOR  0037 A RDR 00000006 001 NONE 08/29 15:08:52                    OPERATOR
U1        0043 A PUN 00000045 001 NONE 08/03 15:05:53  PROFILE  EXEC     U1
```



SPOOL file Descriptor

SPOOL file Descriptor

SPOOL file Descriptor

SPOOL file Contents

SPOOL file Contents

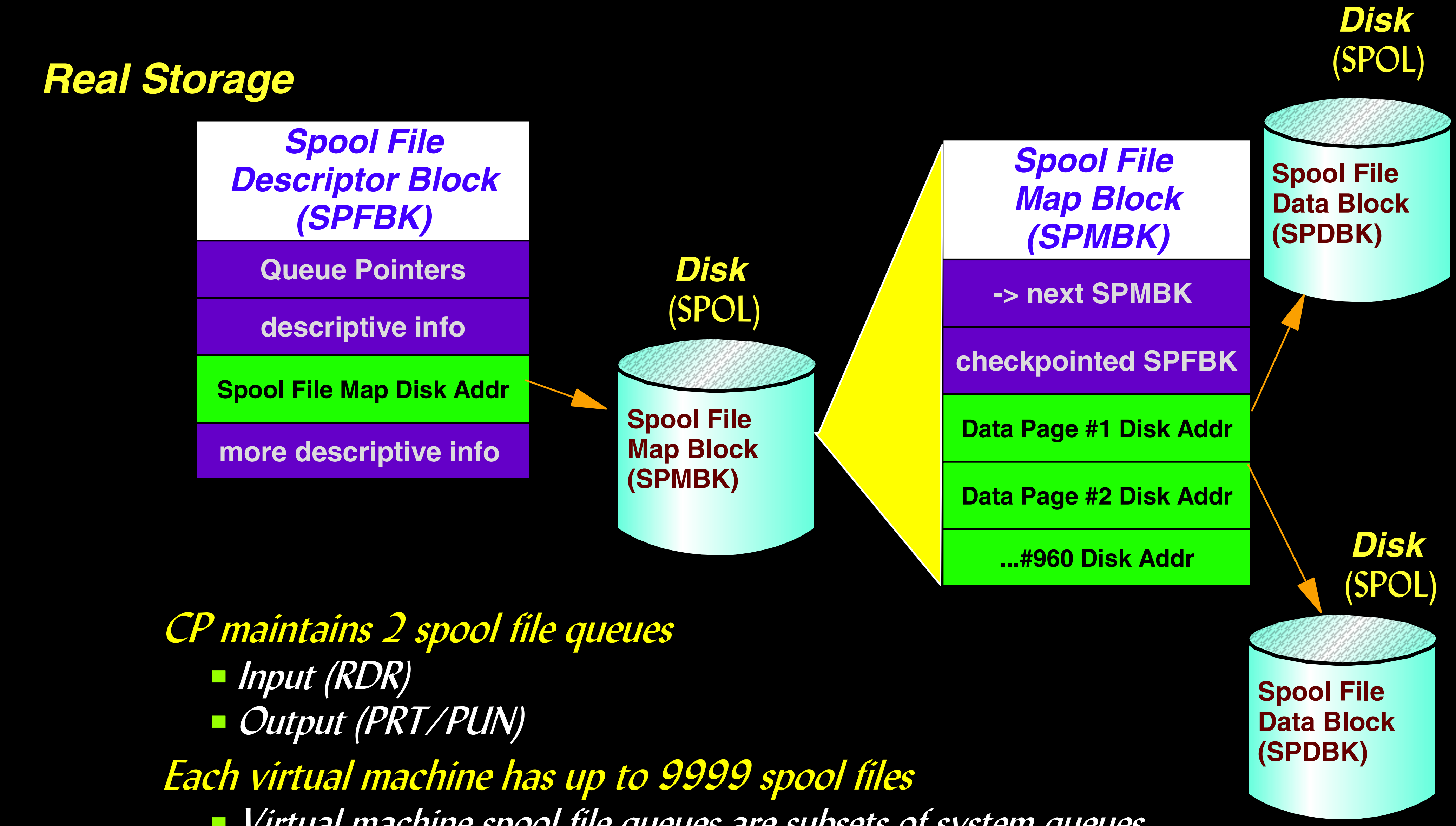
SPOOL file Contents

Real Storage

Disk

(CP-Owned SPOL area)

SPOOL File Structure and Management



CP maintains 2 spool file queues

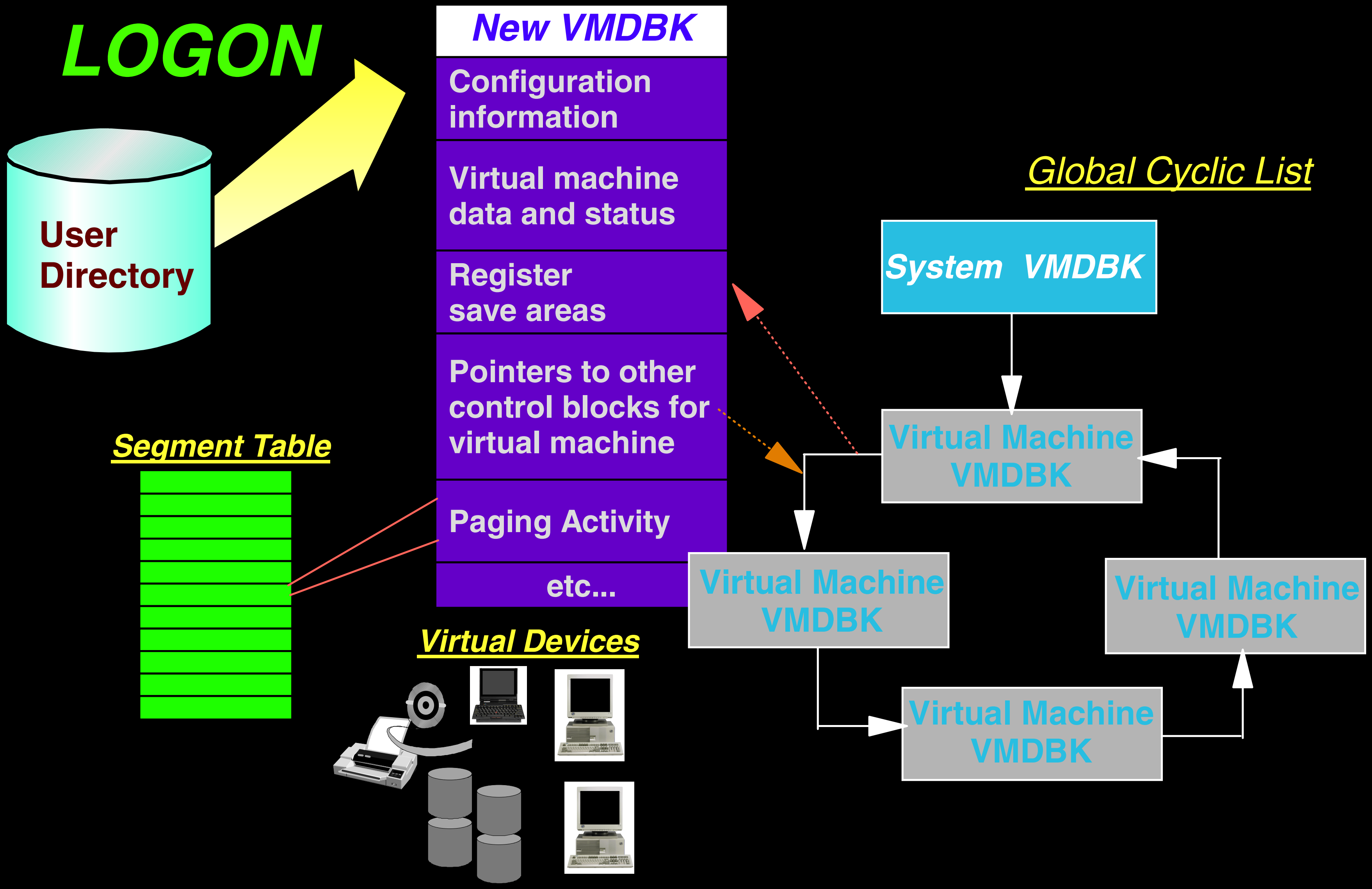
- *Input (RDR)*
- *Output (PRT/PUN)*

Each virtual machine has up to 9999 spool files

- *Virtual machine spool file queues are subsets of system queues*

Running Virtual Machines

Creating a Virtual Machine



How CP Runs Virtual Machines

Virtual machines run in interpretive execution mode

- processes most instructions
- handles Dynamic Address Translation for the virtual machine

CP issues SIE (Start Interpretive Execution) instruction to run a virtual machine

- CP intervention not required until an interrupt or intercept occurs

How CP Runs Virtual Machines ...

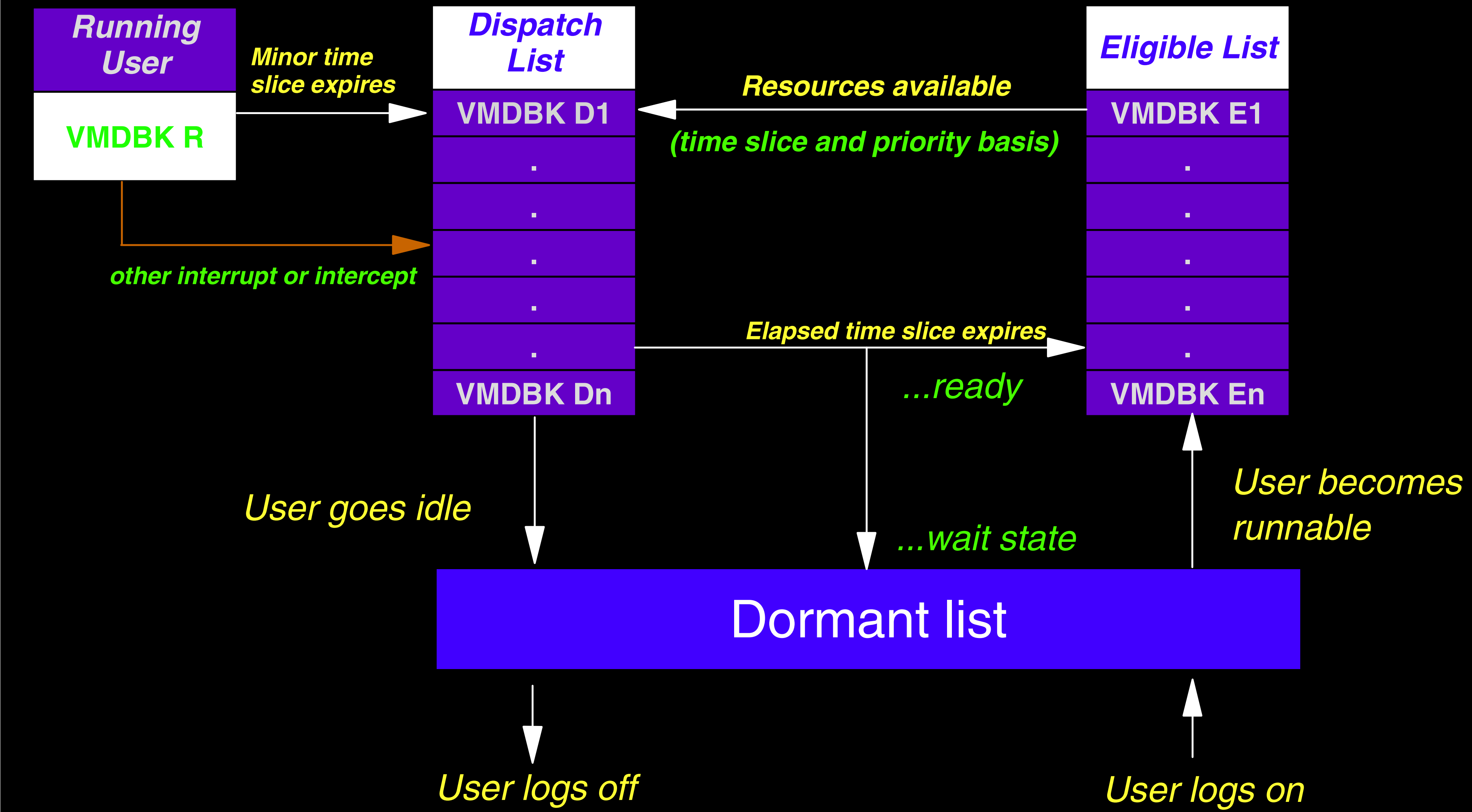
Interrupts

- established time slice expires
- page fault
- I/O operation completes

Intercepts

- interpretive execution facility cannot process an instruction
  - *CP simulates the instruction*
- CP chooses to intercept the instruction
  - *TRACE command targets*
  - *I/O instructions*

Scheduling Virtual Machines to do Their Work



Running User

VMDBK R

Minor time slice expires

other interrupt or intercept

Dispatch List

VMDBK D1

.

.

.

.

.

VMDBK Dn

Resources available

(time slice and priority basis)

Eligible List

VMDBK E1

.

.

.

.

.

VMDBK En

Elapsed time slice expires

...ready

User goes idle

...wait state

User becomes runnable

Dormant list

User logs off

User logs on

Who's Running on My System?

```
ind q
VMLINUX1      Q3 R03 00039068/00039068 JFRANCIS      Q1 R00 00000759/00000739
TCPIP         Q0 EX  00011500/00011483 CORAKR        Q3 IO  00004038/00003909
EDLWRK5       Q3 AP  00002628/00002454 EDLWRK3       Q3 EX  00001720/00001672
DCEPKBLD      Q3 PS  00104747/00104742 EDLWRK1       Q3 AP  00002628/00002259
HUGENBRU      Q3 TI  00002105/00002920 PVMG          Q0 PS  00000237/00000215
VTAM          Q0 PS  00001872/00001728 CORAK2        Q3 PS  00008936/00008936
DSSERV        Q0 PS  00005767/00005766 PVM           Q0 PS  00000629/00000545
VMLINUX       Q3 PS  00003196/00003192 EDLSFS1       Q0 PS  00007770/00007767
Ready;
```

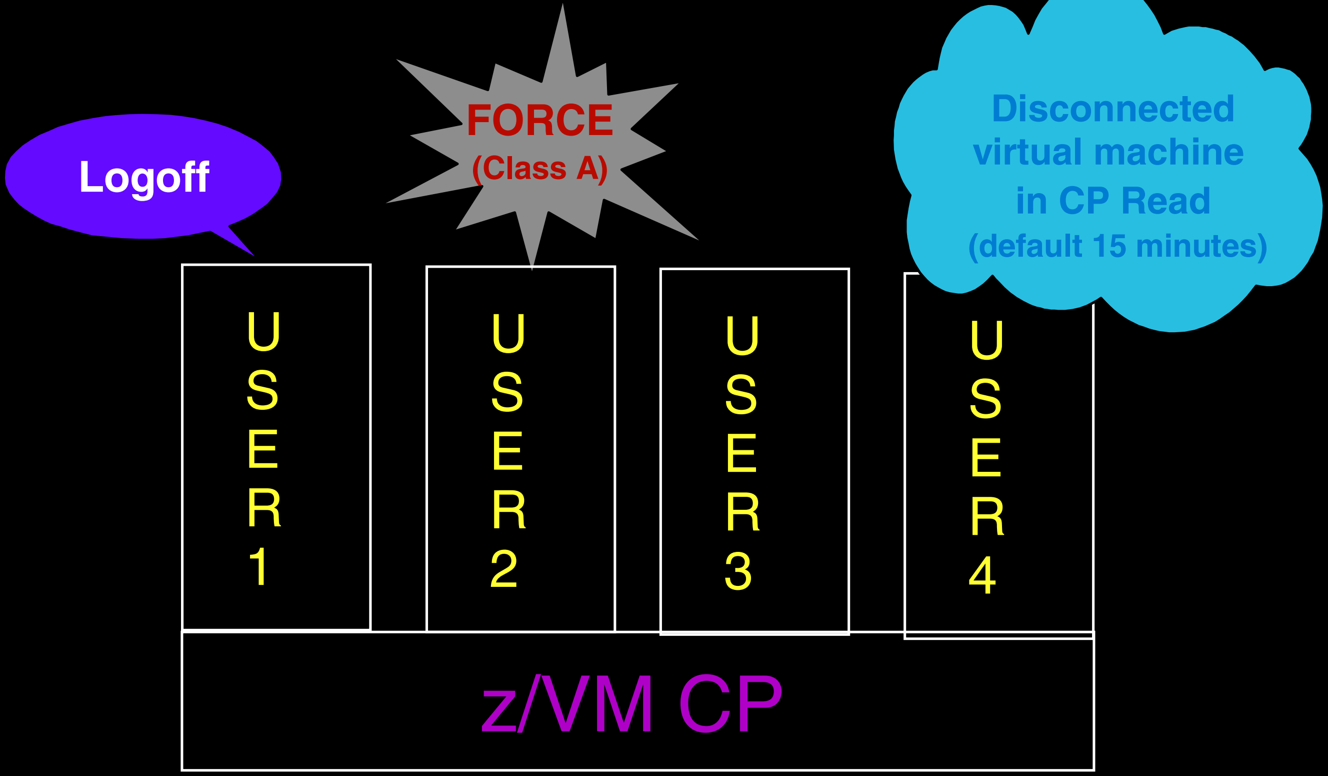
User Transaction Classes

- 0 - special class; never wait in eligible list
- 1 - "short running" transactions
- 2 - "medium running" transactions
- 3 - "long running" transactions

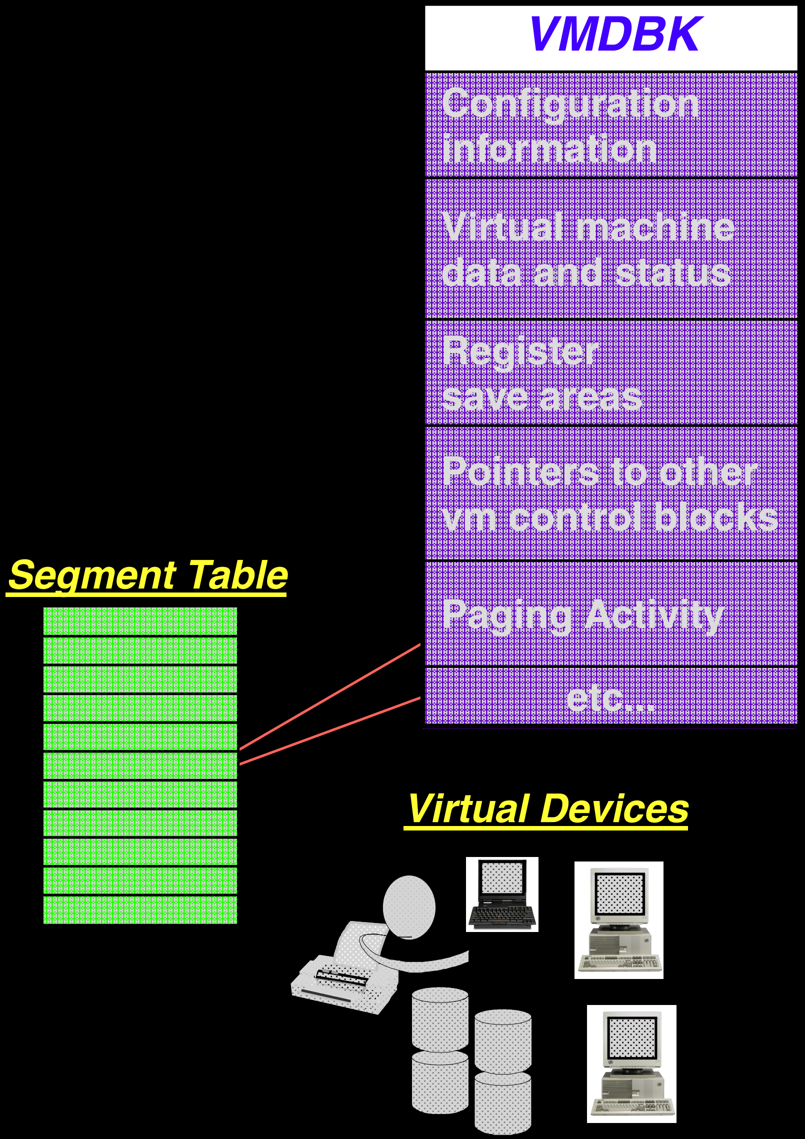
Status Indicators

- Rnn - current RUNUSER on real processor
- EX - instruction simulation wait
- AP - waiting for APPC/VM function
- PS - PSW wait
- TI - test-idle state
- IO - I/O wait
- PG - page wait
- R - ready

How Does a Virtual Machine LOGOFF?



Logging Off a Virtual Machine...



VMDBK

Configuration information

Virtual machine data and status

Register save areas

Pointers to other vm control blocks

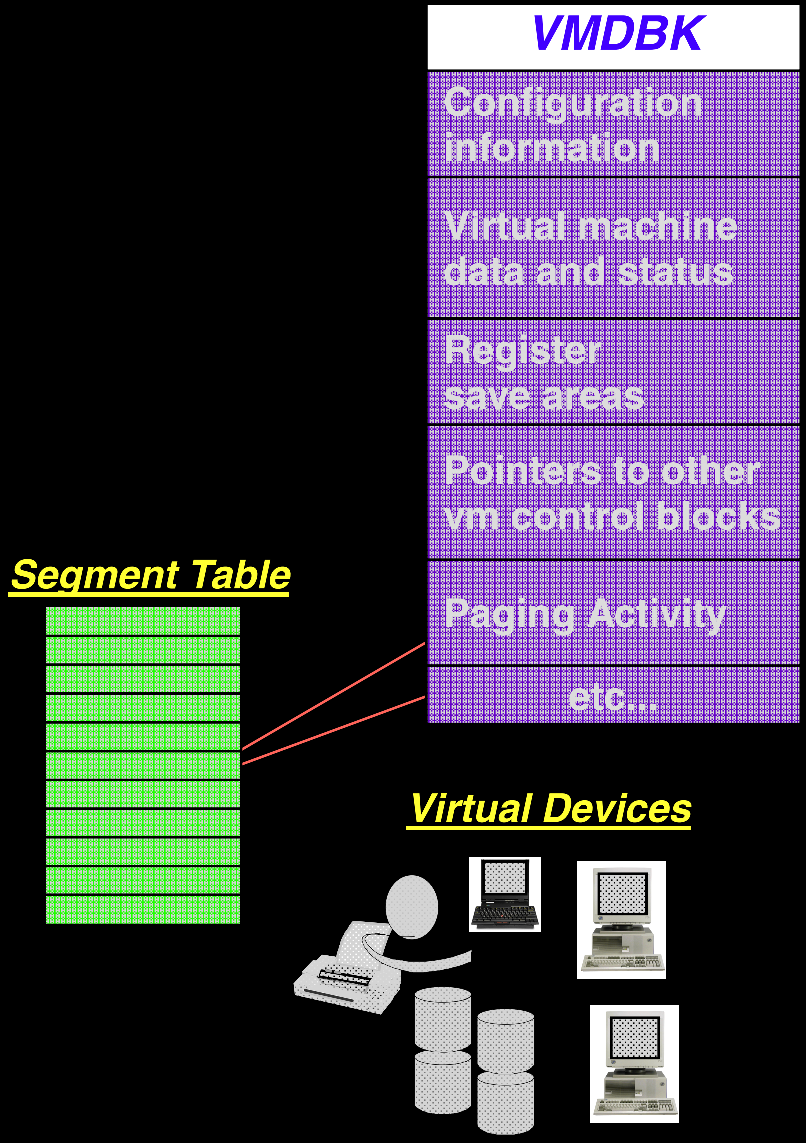
Paging Activity

etc...

Segment Table

Virtual Devices

Logging Off a Virtual Machine...



VMDBK

Configuration information

Virtual machine data and status

Register save areas

Pointers to other vm control blocks

Paging Activity

etc...

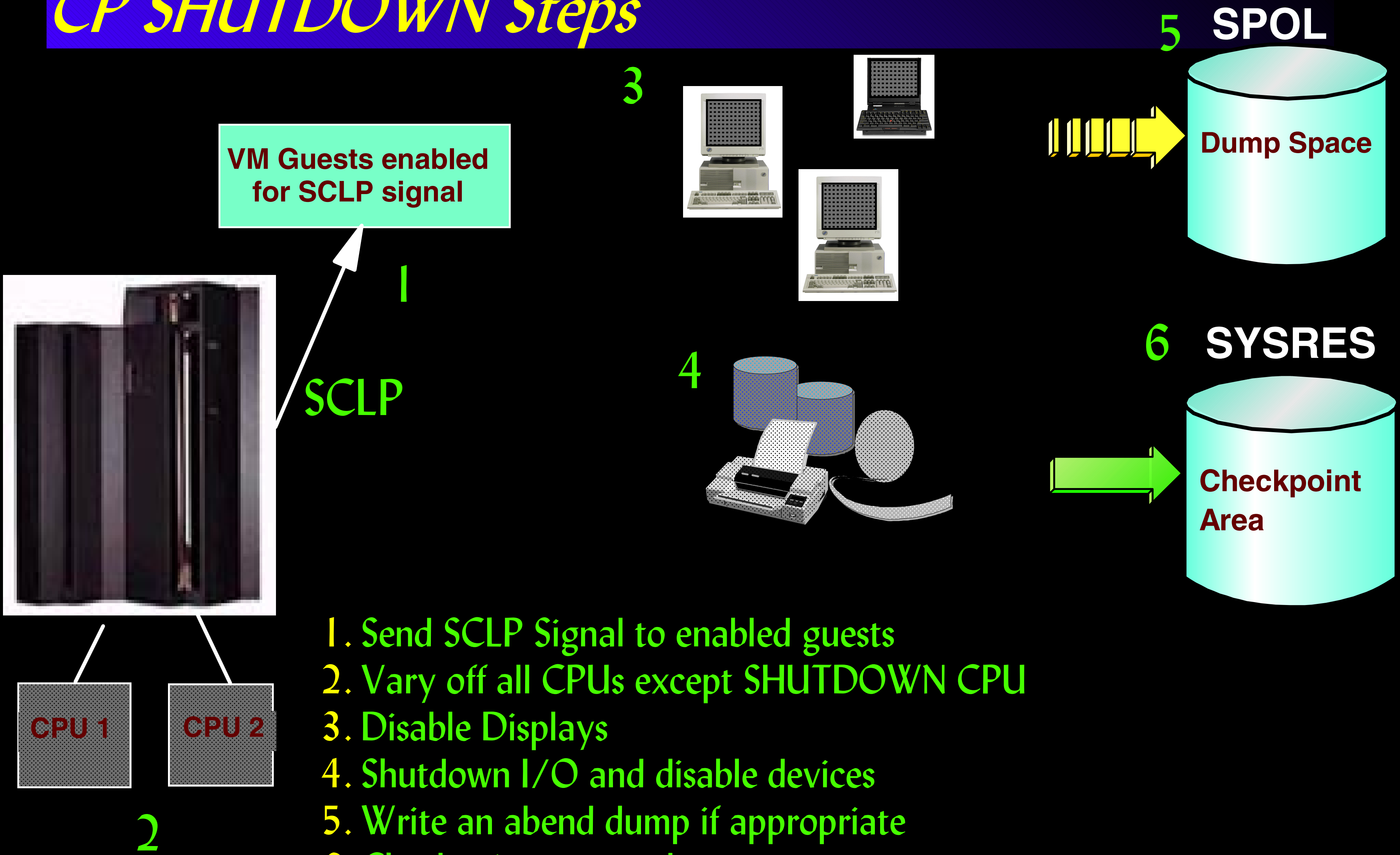
Segment Table

Virtual Devices

1. Complete all outstanding work for virtual machine
   - Close SPOOL files
   - Complete queued tasks
2. Release T-disk assigned to virtual machine
3. Release I/O devices attached to virtual machine
4. Return storage used for virtual machine pages
5. Release virtual machine's control blocks in real storage

Shutting Down CP

CP SHUTDOWN Steps



1. Send SCLP Signal to enabled guests
2. Vary off all CPUs except SHUTDOWN CPU
3. Disable Displays
4. Shutdown I/O and disable devices
5. Write an abend dump if appropriate
6. Checkpoint system data
7. *If REIPL or abend, restart CP*
   - *Otherwise, load disabled wait state and stop*

Collecting Diagnostic Data

Diagnostic Data

Several types of data created by CP can help diagnose problems

- Console messages and logs
- Dumps
  - *System (CP)*
  - *Virtual Machine*
- TRACE Data
- Performance Data
  - *Reports from Performance Tools*
  - *INDICATE commands*
  - *MONITOR data*

Diagnostic Data . . .

Commands may be used to collect additional information

- QUERY
- LOCATE
  - *(5.2.0) : Host Logical and/or Host Real addresses, depending on resource being located*
- DISPLAY
  - *(5.2.0) : Specify Host Logical or Host Real addresses to be displayed*
- etc...

Console Messages and Logs

Most applications and system functions write messages to the virtual machine's console

- System messages are displayed on the operator's console

Console information can be easily saved for review

- SPOOL CONSOLE START command
  - *Begin collecting console data*
  - *Direct console file to desired virtual machine*
- SPOOL CONSOLE STOP/CLOSE command
  - *Stop collecting console data*
  - *Close the file so it may be saved and reviewed*
- RECEIVE file to disk or PEEK it in RDR
  - *Use "(FOR * " if PEEKing file*

CP Dumps

Written to SPOOL or tape

- Determined by SET DUMP command
  - *SET DUMP DASD for SPOOL*

Hard Abend

- Contains all of CP-owned storage

Soft Abend

- Does not cause system termination
- Contains
  - *VMDBK of the active virtual machine at time of abend*
  - *CP Trace Table for processor where error occurred*

SNAPDUMP

- Contains same information as Hard Abend dump
- Does not terminate the system

Other information common to all above dumps

More Dumps

VMDUMP (Virtual Machine Dump)

- Created with VMDUMP command
  - *Unformatted dump*
    - 4K pages of virtual machine's storage
  - *Placed in virtual reader*
  - *DUMPLOAD command used to load into CMS file*

Stand-Alone Dump

- Same format as abend dump
  - *Writes dump of all of main storage*
- Created when stand-alone dump utility is IPLed
  - *Utility created by HCPSADMP EXEC*
    - placed on volume that can be IPLed to start Stand-Alone Dump
- Always written to tape

Processing CP Dumps

CP Dumps are generally sent to OPERATNS reader (RDR)

- DUMPLOAD command processes dumps from RDR (or tape) to disk

The VM Dump Tool is used to analyze dumps

- CP Abend, SNAPDUMP, or Stand-Alone dumps
- Issue VMDUMPTL command

```
z/VM Version 4 Release 4.0, service level 0401 (CP 64-BIT)

Summary of CP exits
      8 Pre-defined exits found
      9 Dynamic exits found
      0 Diagnose exits found

SVC002 (Hard Abend) A restart interrupt occurred.  For a first level
       system, a restart interrupt occurs when the primary system
       operator selects the restart function on the hardware console.
       For a second level system, a restart interrupt occurs when the
       "SYSTEM RESTART" command is entered on the first level console.

Generated at 03/16/04 12:26:58.000000, IPLd at 05/29/04 10:26:05.952420
Date 06/07/04 Time 07:13:33.479393

CPUID = 00097910 20640000

CPU address is 0000    Prefix register is 00024000   (failing)
CPU address is 0001    Prefix register is 7D67C000
CPU address is 0002    Prefix register is 7A8A0000
7E5CA440 07:13:33 Call from HCPRST+594 to HCPDSBOW sav 2275ED00
```

VMDUMPTL - Display Symptom Information

```
>>> symptom
Symptom Record for Incident BB553A5D D61E0SYM

TOD Clock . . BB553A5DD61E0DA0         Date. . . . . 06/07/04
Time Zone . . -04.00.00                Time. . . . . 07:13:33.479393
CPU model . . 2064                     Base SCP. . . 5739
CPU Serial. . 097910                   NodeID. . . . CARVM4
Dump Name . . PMR80417 DUMP0001 01     Dump Type . . CPDUMP
Comp ID . . . 5739A03                  Ver/Rel/Mod . V04R04M0
Dump format . 64-BIT
------------------------------------------------------------
Primary Symptom Strings
              PIDS/5739A0302           (Component ID)
              AB/SSVC002               (Abend Code)
              REGS/FFFFF               (Register/PSW Info)
------------------------------------------------------------
Section 5 Data:
              USERID DUMPED: SYSTEM
              DUMP RECEIVER: OPERATNS
              SPOOLID: 0005
------------------------------------------------------------
Last trace entry on abending processor
7F5CA440 07:13:33 Call from HCPRST+594 to HCPDSBOW sav 2275ED00
                  parm 7D3F4658
------------------------------------------------------------
Abend  Description
...
```

VMDUMPTL - Display Registers

```
>>> regs
R0 1E000B0A_00000000                 R8 00000000_00000000
R1 00000000_2275ED00                 R9 00000000_804B44A8 HCPRST+4A8
R2 00000000_7D3F4658                 RA 00000000_7C43E000
R3 00000000_00000000                 RB 00000000_7C43E000
R4 00000000_7C43E000                 RC 00000000_001B7FB0 HCPSTK
R5 00000000_7ED378F8                 RD 00000000_00026780
R6 00000000_7DDACB50                 RE 00000000_8009DD2A HCPDSB+1FA
R7 00000000_804B4406 HCPRST+406      RF 00000000_001B8130 HCPSTKCP

C0-3 00000000_1400FE40 00000000_00000020 00000000_7F5910C0 00000000_00000000
C4-7 00000000_00000000 00000000_7F591080 00000000_80000000 00000000_7EB88101
C8-B 00000000_00004000 00000000_00000000 00000000_00000000 00000000_00000000
CC-F 00000000_7F5CA461 00000000_00504101 00000000_1F000000 00000000_00000000
...
PFX VALUE 00024000

Restart  Old 04042000 80000000 00000000 001B8156 HCPSTK+1A6
         New 00F43000 80000000 00000000 001BD0D0 HCPSVFDU
External Old 07041000 80000000 00000000 000A13DE HCPDSP+DE
         New 00040000 80000000 00000000 000BA578 HCPEXTEX
SVC      Old 04042000 80000000 00000000 0018578C HCPPGV+2DC
         New 00040000 80000000 00000000 001BCF68 HCPSVFD0
Program  Old 04042000 80000000 00000000 001B81B8 HCPSTK+208
         New 00040000 80000000 00000000 00187E08 HCPPRGIN
Machine  Old 04041000 80000000 00000000 001BB924 HCPSVC+694
```

Tracing

General CP Tracing

- CP builds trace tables for each CPU during initialization
- All occurrences of traceable system events are recorded

VMDUMPTL Display of CP Trace Table

```
>>> TRACE MERGE FOR 100 ONE
7A87B8A0 CPU 0001 /Emerg Signal Ext Int from CPU 0000 parm 00000000
7A88EBA0 CPU 0002 /Emerg Signal Ext Int from CPU 0000 parm 00016000
7F5CA440 CPU 0000 Call from HCPRST+594 to HCPDSBOW sav 2275ED00
7F5CA420 CPU 0000 Return to HCPRST+594 fr HCPDSB+2BC sav 2275ED00
7F5CA400 CPU 0000 Unstack CPEBK at 2275ED00 user MONWRITE retc=0
7F5CA3E0 CPU 0000 Exit to dispatcher from HCPDSB+202 userid MONWRITE
7F5CA3C0 CPU 0000 Stack CPEBK at 2275ED00 user MONWRITE from HCPDSB+1FA
7F5CA3A0 CPU 0000 Call from HCPRST+594 to HCPDSBOW sav 2275ED00
7F5CA380 CPU 0000 Return to HCPRST+594 fr HCPDSB+2BC sav 2275ED00
7F5CA360 CPU 0000 Unstack CPEBK at 2275ED00 user MONWRITE retc=0
7F5CA340 CPU 0000 Exit to dispatcher from HCPDSB+202 userid MONWRITE
7F5CA320 CPU 0000 Stack CPEBK at 2275ED00 user MONWRITE from HCPDSB+1FA
7F5CA300 CPU 0000 Call from HCPRST+594 to HCPDSBOW sav 2275ED00
7F5CA2E0 CPU 0000 Return to HCPRST+594 fr HCPDSB+2BC sav 2275ED00
7F5CA2C0 CPU 0000 Unstack CPEBK at 2275ED00 user MONWRITE retc=0
7F5CA2A0 CPU 0000 Exit to dispatcher from HCPDSB+202 userid MONWRITE
7F5CA280 CPU 0000 Stack CPEBK at 2275ED00 user MONWRITE from HCPDSB+1FA
7F5CA260 CPU 0000 Call from HCPRST+594 to HCPDSBOW sav 2275ED00
7F5CA240 CPU 0000 Return to HCPRST+594 fr HCPDSB+2BC sav 2275ED00
...
```

Tracing...

TRACE Command

- Monitors events in virtual machines
  - Execution of instructions
  - Storage Alteration
  - Register Alteration
  - I/O Activity

Data, I/O, and Guest Tracing

- TRSOURCE and TRSAVE commands
- Data written to system Trace File (TRF)

```
CP TRSOURCE ID TRAP1 SET TRSAMPLE TYPE DATA LOC HCPSPX + C42 41200074
CP TRSOURCE ID TRAP1 SET TRSAMPLE TYPE DATA DL G0:15=REGS
CP TRSOURCE ID TRAP1 SET TRSAMPLE TYPE DATA DL G5.D0=SPFBK

CP TRSAVE FOR ID TRAP1 DASD TO * SIZE 256 KEEP 4

CP TRSOURCE ENABLE SET TRSAMPLE

CP TRSOURCE DISABLE SET TRSAMPLE
QUERY TRF ALL
TRACERED x x x x CMS TRSDATA   OUTPUT A
     where x = spoolid(s) of TRF file(s)
```

Summary

Summary

VM's Control Program (CP):

- Efficiently manages the environment it is running in
  - *Native*
  - *LPAR*
  - *Virtual Machine*
- Preserves and restores data across system IPLs
- Manages processors, memory, and devices among virtual machines
  - *Efficiently shares available resources to meet virtual machine requirements*
- Provides Diagnostic Information
  - *Several types of data*
  - *Many ways to collect it*

See the VM Library for more details
http://www.vm.ibm.com/library/

RETURN TO INDEX