Diagnosing Problems in a UNIX System Services Environment

Session 2920 – Wednesday, February 27, 11:00 AM

John C. Shebey III
IBM Poughkeepsie
jshebey@us.ibm.com

Trademarks

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

- **z/OS®**
- **MVS**
- **OS/390®**

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
UNIX is a registered trademark of The Open Group in the United States and other countries.
SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

* All other products may be trademarks or registered trademarks of their respective companies.

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.

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.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Agenda

- USS Overview
  - Basic Terminology, Environment
  - Shared FS Overview, Common INET
  - Dataspaces, Latches
  - Auxiliary Address Spaces, Product Interaction
  - Syscalls
- Messages and Codes
  - Return Codes, Reason Codes
  - Messages
- Console Commands
- Doc Collection Procedures
  - Failures, Hangs, Loops, Zombies

Agenda (cont)

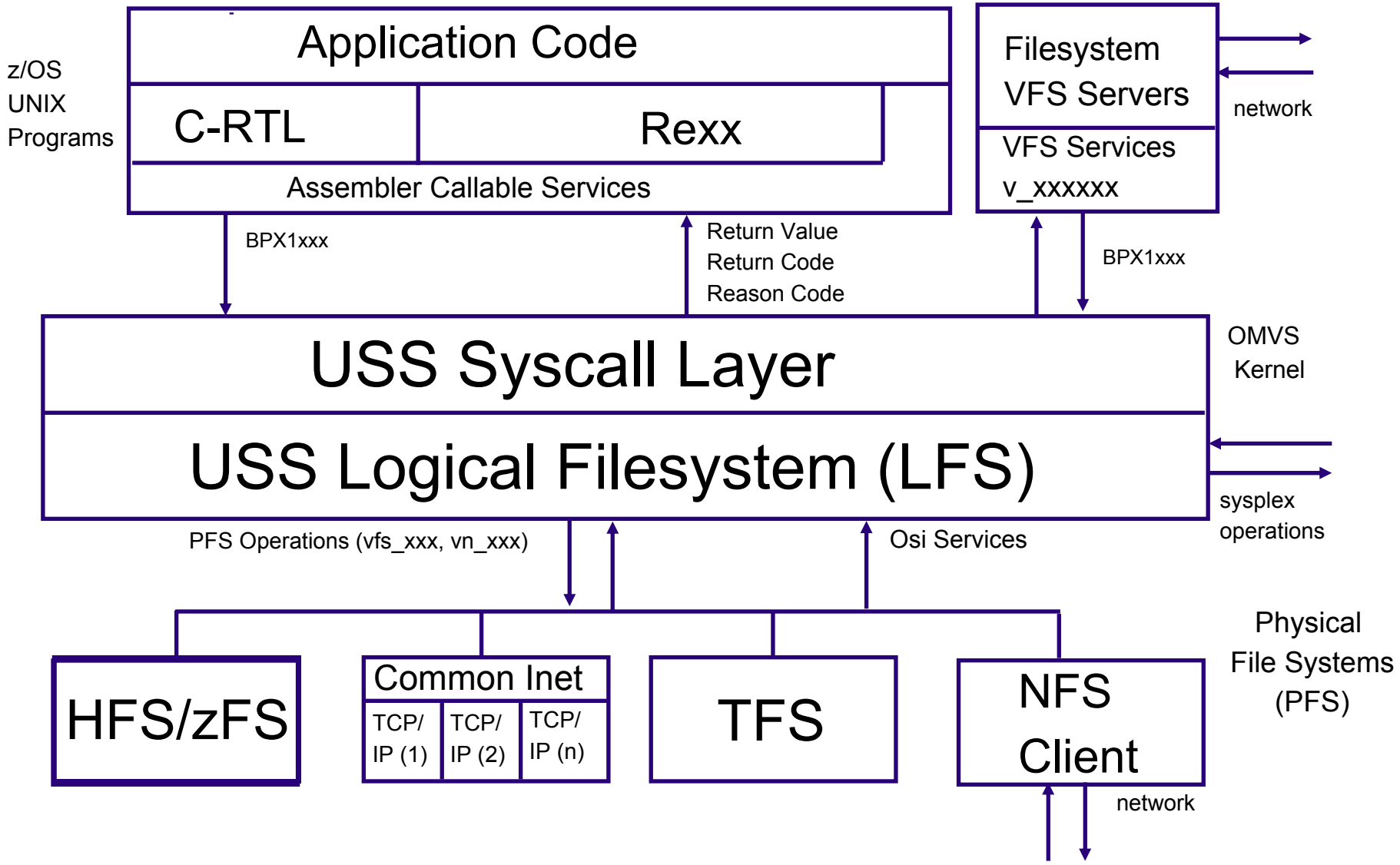
- Documentation
  - Console Dumps, Reason Code SLIP
  - CTRACE
- IPCS Diagnostics
- Problem Scenarios
  - Hard Failures
  - Hangs
- Reference Information
- Appendices
  - Syscall Table
  - Signal Table
  - F BPXOINIT Functions
  - F OMVS Functions
  - OMVS CTRACE to External Writer

USS Overview

Basic Terminology

- **USS** – z/OS UNIX System Services
- **Process** – program using UNIX system services
- **Thread** – a unit (task) of work in a process
- **Dub** – establish a z/OS UNIX environment for an address space
  - create process
- **Fork/Spawn** – methods by which a parent process creates a child process
- **Syscall / System Call / Callable Service**
  - a request by an active process for a service to be performed by z/OS UNIX System Services
- **Zombie** – address space with OMVS resources remaining after dubbed process terminates

z/OS UNIX Environment



Shared File System (Shared FS) Overview

- Ability to share file system data across sysplex
  - Messaging protocol (via XCF services) used to transfer data around sysplex from owner and client systems
- Common File System Hierarchy for all systems in Couple Data Set (CDS) - type **BPXMCDS**
  - Mount and unmount requests are sysplex-wide
- File system availability
  - Recovery in place for maintaining file system availability. Dead system recovery is provided to dynamically recover file systems owned by a system that has left the sysplex for any reason.
- Latches serialize file system operations and I/O to CDS
  - Although latches are system-specific, they can lead to sysplex-wide hangs in a shared FS environment
- CDS serializes shared FS activity across the sysplex

USS Dataspace Usage

- USS uses many dataspaces; some of the ones of more common interest are:
  - SYSZBPX1 - Kernel data
  - SYSZBPXX - CTRACE buffers
  - SYSZBPX2 - File system data
  - SYSZBPX3 - Pipes
  - SYSZBPXC - Common INET sockets
  - SYSZBPXU - AF_UNIX sockets
  - BPXDMxxx - Memory Map
  - BPXDSxxx – Shared Memory
  - BPXFSCDS - FileSys I/O buffer for BPXFSCDS record in OMVS CDS (only used for Shared FS)
  - HFSDSPxx - DFSMS HFS control block caching

USS Latch Sets

- Latches are system specific, and each system has its own set of latches
- USS latch set (LSET) control blocks reside in OMVS private storage
  - **SYS.BPX.A000.FSLIT.FILESYS.LSN**
    - File system latch set - serialize mount, unmount, quiesce, and other file system operations
    - Latch#2 = MOUNT latch
    - Latch#8…n = File System or Quiesced File System latch (prior to z/OS R9)
  - **SYS.BPX.A000.FSLIT.QUIESCE.LSN**
    - Quiesce latch set created separately (z/OS R9)
  - **SYS.BPX.A000.FSLIT.FILESYS.LSN.01 (02,03,...,0n)**
    - File latch sets – serialize file operations
  - **SYS.BPX.A000.PRTB1.PPRA.LSN**
    - Process latch set - serialize process termination and thread operations
- Refer to z/OS R8 MVS Diagnosis: Reference, Understanding UNIX System Services latch contention

Auxiliary Address Spaces

- **BPXOINIT**
  - Runs /etc/init (/etc/rc) during OMVS initialization
    - Initialization script
  - Handles MODIFY BPXOINIT console commands
  - Zombie cleanup
- **BPXAS**
  - Used when new address space is needed for fork processing
  - WLM managed
  - Remain active for 30 minutes after last use
- **Colony Address Spaces**
  - Address spaces in which Physical File Systems can be initialized

Product Interaction

- LE C/C++ RTL to support POSIX API
- TSO/ISPF
  - OMVS shell and ISHELL
- RACF or OEM for security
- TCP/IP for network access
- WLM for fork initiator management (BPXAS)
- GRS for latch operations
- XCF for cross system requests (shared FS sysplex)
- Various physical file systems (HFS, zFS, DFS, NFS,TFS, XPFS,TCP/IP)
- Major e-Business applications
  - Lotus Domino, Websphere, DB2, JAVA, CICS, IMS

z/OS USS Syscalls

- USS API provided by set of Syscalls
  - AMODE 31 – BPX1xxx or BPX2xxx
  - AMODE 64 – BPX4xxx
    - z/OS R6 and higher release levels
- Documented in z/OS USS Assembler Callable Services Reference
  - Function
  - Format
  - Parameters
    - Function specific parms
    - Return Value - (rv)
    - Return Code - (errno)
    - Reason Code - (errnojr)
  - Usage Notes
- Note: See ***Appendix A*** for Syscall Numbers

Syscall Return Information

- Syscall Return Information
  - The return value (rv) is set to -1 (x'FFFFFFFF') if an error condition is found
  - The return code (errno) contains an integer value associated with a generic error condition
  - The reason code (errnojr) contains a four-byte value and it provides specific information about the failure
- In general, messages or traces contain the corresponding return value, return code and reason code

Messages and Codes

USS Return/Reason Codes

- Refer to z/OS USS Messages and Codes
- USS Return Codes (errno) generally correspond to standard POSIX errors, i.e. EAGAIN, EACCES, EBUSY
- z/OS UNIX Reason Codes (errnojr)
  - A full word (CCCCRRRR)
  - CCCC is the reason code qualifier - component
  - RRRR is the reason code
  - If CCCC is between 0-20FF then RRRR is a USS reason code
    - Use **BPXMTEXT** from TSO, IPCS or Shell
- Reason codes from other components
  - See z/OS USS Messages and Codes: Reason Codes (Errnojrs) Listed by Value

Reason Code Ranges for Other Components

- 5B00 - 5BFF
  - DFSMS HFS File System
- 6C00-6CFF
  - Distributed File Systems Client (DFSC)
- 6E00-6EFF
  - NFS Client File System
- 7100-71FF
  - VTAM Anynet Sockets
- 7300-73FF
  - z/OS Communications Server TCPIP Stack
- 7880-78FF
  - z/OS Communications Server Resolver
- EF01-EFFF
  - zFS File System

Example Usage: BPXMTEXT

- BPXMTEXT is a tool that can be used to quickly interpret a USS errnojr (reason code qualifier between 0-20FF)
  - Shipped in SYS1.SBPXEXEC
    - Must be in SYSEXEC or SYSPROC to use
  - z/OS R8, also interprets zFS (EFxx) and TCP/IP (7xxx) reason codes
  - z/OS R9, also interprets C/C++ run-time library (Cxxx) reason codes
- USAGE: BPXMTEXT errnojr
  - Use as either a TSO, Shell, or IPCS command
- TSO BPXMTEXT 058800AA

BPXFSUMT 09/24/04
JRFsParentFs: The file system has file systems mounted on it

Action: An unmount request can be honored only if there are no file systems mounted anywhere on the requested file system. Use the D OMVS,FILE command from the system console to find out which file systems are mounted on the requested file system. Unmount them before retrying this request.

ETCINIT Exit Status Codes

- Used by /usr/sbin/init (/etc/init) to report a failure during initialization to operator console
  - BPXI027I THE ETCINIT JOB ENDED IN ERROR, EXIT STATUS 0000xx00
    - Refer to z/OS USS Messages and Codes: Exit Status Codes for /usr/sbin/init
  - For Exit Status of the form 000000xx, xx is the signal number of the signal that ended the /usr/sbin/init process
    - Refer to z/OS USS Programming: Assembler Callable Services Reference: Signal defaults
    - Note: See also ***Appendix B*** for Signal Numbers

USS Abend Codes

- Refer to z/OS MVS System Codes
- Hexadecimal reason code in register 15 describes the error
- SEC6 (abendEC6)
  - Reason Code: CCCC RRRR
  - RRRR values
    - For example: FDxx ,FFxx
    - If xx is in the range of x'01' to x'7F', a signal was received
- S422 (abend422)
  - USS Reason Code: xxxx01zz
    - If zz is in the range of x'01' to x'7F', a signal was received

USS Related Messages

- BPX - z/OS USS Operator Console Messages
- FSUM - Shell & Utilities messages
- FDBX - Debugger (DBX) messages
- FOM - Application Services messages
- EDC - LE C/C++ RTL messages
- CEE - LE base messages
- CBC - C/C++ Compiler messages
- EZx - TCP/IP messages
- ICH/IRR - RACF messages

USS Related Messages (cont)

- IMW - WebSphere messages
- IGW - DFSMS/HFS messages
- IOE - DFS/zFS messages

- **LOOKAT** Message Tool
  - http://www-1.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/

Console Commands

USS Display Commands

- See command response message in z/OS MVS System Messages for more details
  - D OMVS - OMVS kernel status
    - BPXO042I - Parmlib member, Status
  - D OMVS,A=ALL - lists all dubbed processes
    - BPXO040I -Jobname, PID, PPID, ASID, State, CMD
  - D OMVS,P - displays Physical File System information
    - BPXO046I - Type, Description, Entry, Maxsock
  - D OMVS,F - produces a mounted file system report
    - BPXO045I - Path, Name, Mode, Typename, Status, Latch QSYSTEM, QPID, QJOBNAME, Owner, Automove, Client

USS Display Commands (cont)

- D OMVS,O - lists BPXPRMxx settings
  - BPXO043I
    - SYSPLEX=YES -> Shared FS
- D OMVS,L - System Wide Limits
  - BPXO051I - Current Usage, Highwater Usage, System Limit, LIMMSG
- D OMVS,L,PID=xxxx - Process Limit
  - BPXO051I - Process Information, Limits
- D OMVS,PID=xxx - Displays thread information for process
  - BPXO040I - Process Information, Thread Information (for each thread: Thread ID, TCB, Syscall, State)
- D GRS,C – Displays possible ENQ and USS latch contention
  - ISG343I

USS Display Commands
D OMVS,A=ALL

- See command response message in z/OS MVS System Messages for more details
  - D OMVS - OMVS kernel status
    - BPXO042I - Parmlib member, Status
  - **D OMVS,A=ALL** - lists all dubbed processes
    - BPXO040I -Jobname, PID, PPID, ASID, State, CMD
    - Can be used to display currently active USS processes when diagnosing loops/high CPU in USS environment
    - Can also be used for identifying zombie buildup
  - D OMVS,P - displays Physical Filesystem information
    - BPXO046I - Type, Description, Entry, Maxsock
  - D OMVS,F - produces a mounted file system report
    - BPXO045I - Path, Name, Mode, Typename, Status, Latch
      QSYSTEM, QPID, QJOBNAME, Owner, Automove, Client

USS Display Commands
D OMVS,F

- See command response message in z/OS MVS System Messages for more details
  - D OMVS - OMVS kernel status
    - BPXO042I - Parmlib member, Status
  - D OMVS,A=ALL - lists all dubbed processes
    - BPXO040I -Jobname, PID, PPID, ASID, State, CMD
  - D OMVS,P - displays Physical Filesystem information
    - BPXO046I - Type, Description, Entry, Maxsock
  - **D OMVS,F** - produces a mounted file system report
    - BPXO045I - Path, Name, Mode, Typename, Status, Latch QSYSTEM, QPID, QJOBNAME, Owner, Automove, Client
    - Can be used when diagnosing hang in USS environment to look for quiesced file system
    - Also useful in shared FS environment to compare each system's local view of mounted file systems
    - At z/OS R7, latch numbers displayed for each file system, as well as date and time each file system was mounted

USS Display Commands
D GRS,C

- D OMVS,O - lists BPXPRMxx settings
  - BPXO043I
    - SYSPLEX=YES -> Shared FS
- D OMVS,L - System Wide Limits
  - BPXO051I - Current Usage, Highwater Usage, System Limit, LIMMSG
- D OMVS,L,PID=xxxx - Process Limit
  - BPXO051I - Process Information, Limits
- D OMVS,PID=xxx - Displays thread information for process
  - BPXO040I - Process Information, Thread Information (for each thread: Thread ID, TCB, Syscall, State)
- **D GRS,C**
  - ISG343I - Displays possible ENQ and USS latch contention
  - Normally used to check for latch contention when diagnosing hang in USS environment
  - Can also be used to look for latch or other resource contention when diagnosing loop/high CPU in USS environment

USS Display Commands Added at z/OS R7

- See command response message in z/OS MVS System Messages for more details
  - **D OMVS,MF** - last 10 (or less) move and mount failures
    - BPXO058I
  - **D OMVS,MF=ALL** - last 50 (or less) move and mount failures
    - BPXO058I
  - **D OMVS,W** - MOUNT latch tracking and cross-system messages
    - BPXO063I

    1. Information about holder of the MOUNT latch and waiting tasks, including what operation the holder is doing, how long the latch has been held and how long the waiters have been waiting, and what file system (if any) is involved
    2. Information about shared FS messages that have been sent to another member of the shared FS sysplex for which a response has not yet been received
    3. Information about shared FS messages that have arrived at this system and that have not yet been responded to

USS Display Commands Added at z/OS R8

- **D OMVS,W**
  - Display output was enhanced at z/OS R8
    - BPXO063I

    1. Displays information on File System Latch contention, similar to what is shown for the Mount Latch. Only shows up if waiters exist for a File System Latch.
    2. Displays information on threads waiting for other reasons. This includes threads that have been waiting within a Physical File System for more than 5 seconds (may indicate a problem in the PFS), latch waits for a file, Byte Range (BRLM) Lock waits, and HSM recall within automount for a new mount in progress.
  - APAR **OA17319** rolls this z/OS R8 'D OMVS,W' enhancement back to z/OS R7

F BPXOINIT Shared FS Diagnostics

- F BPXOINIT,FILESYS=D,GLOBAL
  - BPXF041I - Displays current sysplex state, including active systems (system name, LFS version level), and active serialization categories:
    - **SYSTEM PERFORMING INITIALIZATION**
    - **SYSTEM PERFORMING MOVE**
    - **SYSTEM PERFORMING QUIESCE**
    - **SYSTEMS PERFORMING UNMOUNT**
    - **SYSTEMS PERFORMING MOUNT RESYNC**
    - **SYSTEMS PERFORMING LOCAL FILE SYSTEM RECOVERY**
    - **SYSTEMS PERFORMING FILE SYSTEM TAKEOVER RECOVERY**
    - **SYSTEMS RECOVERING UNOWNED FILE SYSTEMS**
    - **SYSTEMS PERFORMING REPAIR UNMOUNT**
  - If possible, issue from system with highest LFS version level that does not have MOUNT Latch contention

F BPXOINIT Shared FS Diagnostics (cont)

- F BPXOINIT,FILESYS=D,FILESYSTEM=file_system_name
  - Displays information about file system from sysplex (CDS) perspective
- F BPXOINIT,FILESYS=D,ALL
  - Displays all file systems in the CDS shared FS hierarchy
- F BPXOINIT,FILESYS=D,EXCEPTION
  - Displays file systems in exception state, including differences between local and sysplex (CDS) view
  - Exception states include: Mount in progress, Unmount in progress, Quiesce in progress, Quiesced, Unowned , In recovery, Unusable
- F BPXOINIT,FILESYS=DUMP
  - Capture dump of OMVS and dataspaces, including CDS sub-records
    - Should not be issued on system with MOUNT Latch contention

F BPXOINIT Shared FS Diagnostics (cont)

- F BPXOINIT,FILESYS=FIX
  - Note: Use with CAUTION and issue from system with highest LFS version, if possible
  - Captures dump of OMVS and dataspaces, including sub-records in the CDS, prior to diagnosis and repair
  - Performs automatic file system and CDS diagnosis and repair
  - Typically used for display purposes to isolate problem system
  - FIX processing has both synchronous processing from the system that the FIX was issued, and also asynchronous processing that occurs on all other systems in the shared FS sysplex
  - FIX issues messages pointing to specific system(s) causing delay
    - **Need to check the hardcopy system logs of all systems in the shared FS sysplex for any messages issued in response to FIX**

F BPXOINIT Shared FS Diagnostics (cont)

- F BPXOINIT,FILESYS=FIX messages
  - Message **BPXF049I** for each file system that is delayed during unmount or quiesce processing. The message also lists the systems that are causing the delay.
  - Message **BPXF042I** for each system that has contention for the file system MOUNT latch. Contention for the MOUNT latch delays high-level functions, such as mount and unmount processing.
  - Message **BPXF057I** for each file system that has latch contention. The message identifies the file system and the system where the latch contention is occurring.
  - Hardcopy message **BPXF048I** for each correction made to the file system global data structures (in the BPXMCDS couple data set).

Latch Recovery Support

- At z/OS R6, support was added to more proactively identify USS latch contention and recover
- Message **BPXM056E** issued automatically to provide notification of z/OS USS latch contention if latch held by same holder for 5 minutes
  - D GRS,C can then be used to show latch contention details
  - BPXM056E will be DOMed if contention is relieved
- F BPXOINIT,RECOVER=LATCHES can aid in relieving contention
  - Abends (422-01A5) **user** address space latch holder tasks
  - Handles case where latch holder is system task that is in OSI_Wait out of PFS
    - Ensure that fix for HIPER APAR **OA12827** is applied
  - Requests system dump to capture potential problem
    - APAR **OA16218** will prevent abend422 rsn1A5 dumps from being suppressed
    - At z/OS R9, zFS address space included in dump, and multi-system dumps taken (shared FS) if MOUNT/VFS latch held by abended task
  - Issues message **BPXM057E** (CONTENTION NOT RESOLVING) or **BPXM067I** (CONTENTION RESOLVED)

Doc Collection Procedures

Doc Collection Procedures – Abends and Failing Reason Codes

- Refer to INFO APAR **II08038** for general doc collection procedures
- **Abends (i.e. abendEC6)**
  - Normally generate SVC dump
    - Do not normally get dumps for abendEC6 rsn0000FFxx
  - If no SVC dump generated, may need to SLIP on abend
  - May need OMVS CTRACE
  - May need SYSLOG and EREP data
- **Failing Reason Code**
  - Often need SLIP dump for reason code
  - May need OMVS trace
  - May need SYSLOG and EREP data

Doc Collection Procedures – Hang in Non-Shared FS Environment

- **Non-Shared FS Hang**
  - Issue D GRS,C
  - If FILESYS latch contention, issue D OMVS,F to check for quiesced file system as possible culprit
    - Message **BPXF034I** issued on system that owns quiesced file system if file system has been quiesced for more than 10 minutes (not seen for zFS file systems)
  - At z/OS R7 and higher releases, issue 'D OMVS,W' if latch contention
    - Look for file system involved and associated PFS
    - z/OS R8 MVS Diagnosis: Reference, Procedure: Diagnosing and resolving latch contention
  - Need console dump of OMVS and its dataspaces, along with:
    - Any non-OMVS latch holders (if USS latch contention)
    - Any hung jobs and/or users
    - Potentially any active PFS address spaces on failing system
  - May need SYSLOG and EREP data
    - SYSLOG would contain output from above display commands

Example: FTPD Hung

- D GRS,C

```
ISG343I 15.28.43 GRS STATUS 066
LATCH SET NAME:  SYS.BPX.A000.FSLIT.FILESYS.LSN
CREATOR JOBNAME: OMVS      CREATOR ASID: 000E
  LATCH NUMBER:  12
    REQUESTOR  ASID  EXC/SHR    OWN/WAIT
    OMVS       000E  EXCLUSIVE  OWN  (2)
    FTPD       003E  EXCLUSIVE  WAIT (1)
  LATCH NUMBER:  93
   REQUESTOR  ASID  EXC/SHR    OWN/WAIT
    WELLIE3   00C2  EXCLUSIVE  OWN  (4)
    OMVS      000E  SHARED     WAIT (3)
```

- (1) FTPD hung waiting for FILESYS Latch#12
- (2) Latch#12 owned by OMVS
- (3) OMVS hung waiting for FILESYS Latch#93
- (4) Latch#93 owned by WELLIE3
- WELLIE3 is at root of contention - include WELLIE3 in dump

Doc Collection Procedures – Loop/ High CPU Non-Shared FS

- **Non-Shared FS Loop/High CPU**
  - Issue D OMVS,A=ALL and D GRS,C
  - Enable OMVS CTRACE
  - Enlarge the system trace table size to its max
    - TRACE ST,999K
  - Need console dump of OMVS and its dataspaces, along with:
    - Any non-OMVS latch holders (if USS latch contention)
    - Any address spaces that are looping or using high CPU
  - May need SYSLOG and EREP data
    - SYSLOG would contain output from above display commands

Doc Collection Procedures – Zombie Buildup

- **Zombie Buildup**
  - Issue D OMVS,A=ALL
    - Zombie processes have a state of **1Z**
  - Do zombies have parent with PID=1 (PPID=1)?
    - If so, issue **D J,BPXOINIT** to verify that BPXOINIT is swapped in and running
    - Need console dump of OMVS and its dataspaces, including BPXOINIT address space
  - If parent of zombies does not have PID=1, then it is responsibility of parent to cleanup zombies
    - Did parent process issue a **waitpid()** to wait for child process to end?

Doc Collection Procedures – OMVS Initialization Hang

- **Hangs During OMVS Initialization**
  - Review SYSLOG and look for message BPXP006E
  - BPXP006E OMVS IS:
    - **INITIALIZING THE FILE SYSTEM**
    - **CREATING THE BPXOINIT ADDRESS SPACE**
    - **PROCESSING IN BPXOINIT**
    - **STARTING THE INITIALIZATION PROCESS**
    - **RUNNING THE INITIALIZATION PROCESS**
    - **WAITING FOR SECURITY PRODUCT INITIALIZATION**
    - **WAITING FOR CATALOG ADDRESS SPACE INITIALIZATION**
    - **WAITING FOR JOB ENTRY SUBSYSTEM INITIALIZATION**
    - **OMVS IS UNABLE TO CREATE THE BPXOINIT ADDRESS SPACE**

Doc Collection Procedures – OMVS Initialization Hang (cont)

- Also review SYSLOG for message BPXP007E
  - **BPXP007E STARTING PHYSICAL FILE SYSTEM** *pfsname* **IN ADDRESS SPACE** *spacename*
- If OMVS is stuck running the initialization process, review **/etc/log** to see output from **/etc/rc** and last command that successfully ran
- A console dump of OMVS and its dataspaces may need to be taken, along with:
  - Any address spaces involved in hang
- EREP data may be needed to look for initialization errors

- Message **BPXI004I OMVS INITIALIZATION COMPLETE** issued when OMVS finishes initializing

Shared FS Basic Diagnosis

- The two goals of Shared FS Diagnosis are:
  - Correcting the problem or limiting the scope of the outage to a single system or a subset of systems
  - Provide enough information about the problem to enable the IBM Support Center to identify and resolve the root cause of the problem
- z/OS MVS Diagnosis: Reference has detailed information about shared FS diagnosis and recovery for various problem scenarios
  - Diagnostic Procedures for Shared File System
  - z/OS R8 - Procedure: Diagnosing and resolving latch contention
- INFO APAR **II08038** further describes shared FS doc collection

Doc Collection Procedures – Hang in Shared FS Environment

- F BPXOINIT,FILESYS=DISPLAY,GLOBAL
  - Identifies active systems (system with highest level LFS code), and shows if system(s) performing serialized activity
- RO *ALL,D GRS,C
  - Look for contention on latch set SYS.BPX.A000.FSLIT.FILESYS.LSN, especially with Latch #2 (MOUNT Latch), and PPRA latch set
- RO *ALL,D OMVS,F
  - Shows a local view of the file system hierarchy for each system
- F BPXOINIT,FILESYS=D,ALL
  - Shows CDS (sysplex) view of file system hierarchy
- RO *ALL,F BPXOINIT,FILESYS=DISPLAY,EXCEPTION
  - Shows differences between local and CDS (sysplex) view of file systems, as well as file system *exception* states
- z/OS R7 and higher releases: RO *ALL,D OMVS,W
  - If MOUNT or File System Latch contention, shows information about holder of latch, including file system involved (if any) and operation; look at how long latch held
  - Review waiting threads not involved in MOUNT or File System Latch contention
  - Also check for any long outstanding cross-system messages

Doc Collection Procedures – Hang in Shared FS Environment (cont)

- Capture EREP data and SYSLOGs for every system in the shared FS sysplex – a merged OPERLOG is preferable
  - The SYSLOGs should contain all of the above information
- Capture sysplex-wide dumps of OMVS and its dataspaces
  - Dump any hung address spaces involved in problem
  - On systems with latch contention, be sure to dump any non-OMVS latch holders separately
  - Also dump any PFS address spaces that may be relevant to the problem (ie. ZFS, NFS, DFSKERN, etc.)
- **If uncertain what doc to collect, contact IBM support immediately!**

Doc Collection Procedures – Hang in Shared FS Environment (cont)

- Review the F BPXOINIT,FILESYS=D,GLOBAL output to identify if any system(s) performing serialized event(s)
  - **SYSTEM PERFORMING INITIALIZATION**
  - **SYSTEM PERFORMING MOVE**
  - **SYSTEM PERFORMING QUIESCE**
  - **SYSTEMS PERFORMING UNMOUNT**
  - **SYSTEMS PERFORMING MOUNT RESYNC**
  - **SYSTEMS PERFORMING LOCAL FILE SYSTEM RECOVERY**
  - **SYSTEMS PERFORMING FILE SYSTEM TAKEOVER RECOVERY**
  - **SYSTEMS RECOVERING UNOWNED FILE SYSTEMS**
  - **SYSTEMS PERFORMING REPAIR UNMOUNT**
- May need to issue F BPXOINIT,FILESYS=FIX to further isolate problem system(s)

Doc Collection Procedures – Hang in Shared FS Environment (cont)

- Check hardcopy log of each system for FIX responses
  - Message **BPXF049I** for each file system that is delayed during unmount or quiesce processing. The message also lists the systems that are causing the delay.
  - Message **BPXF042I** for each system that has contention for the file system MOUNT latch. Contention for the MOUNT latch delays high-level functions, such as mount and unmount processing.
  - Message **BPXF057I** for each file system that has latch contention. The message identifies the file system and the system where the latch contention is occurring.
  - Hardcopy message **BPXF048I** for each correction made to the file system global data structures (in the CDS).

Doc Collection Procedures – Loop/ High CPU in Shared FS Environment

- Issue D OMVS,A=ALL to display currently active USS processes on the system with high CPU or loop
- RO *ALL,D GRS,C
  - Look for latch contention on all systems, especially concerned with FILESYS latch set (Latch#2 – MOUNT Latch) or PPRA latch set
- F BPXOINIT,FILESYS=D,GLOBAL
  - Check for any systems persistently performing serialized activity
- Enable OMVS CTRACE on system with high CPU or loop before taking dump
  - May also want to enable OMVS CTRACE on other systems in shared FS sysplex before taking dumps
- Enlarge the system trace table size on system with high CPU or loop to its max
  - TRACE ST,999K
- Dump OMVS and its dataspaces on all systems, along with:
  - Non-OMVS latch holders (if USS latch contention)
  - Any looping address spaces or address spaces involved with high CPU
- May need SYSLOG and EREP data from all systems

Documentation

Getting a Console Dump

- Console Dump Example:

```
DUMP COMM=(description or dump name)
R xx,JOBNAME=(OMVS,other-related-processes),CONT
R xx,SDATA=(PSA,SQA,LSQA,RGN,TRT,LPA,CSA,SUM,ALLNUC),CONT
R xx,DSPNAME=('OMVS'.*),END
```

- Instead of (or in conjunction with) JOBNAME, ASID=(omvsasid,other_process_asids) may be used
- DSPNAME=('OMVS'.SYS*,'OMVS'.BPX*) will not capture HFS dataspaces and may save a lot of space in the dump dataset
- May need to use wildcard '*' for jobname of forked/ spawned address spaces

Getting Console Dumps on All Systems in Shared FS Sysplex

- Example of getting console dumps on all systems in shared FS sysplex:

```
DUMP COMM=(description or dump name)
JOBNAME=(OMVS),DSPNAME=('OMVS'.*),
SDATA=(CSA,LPA,TRT,RGN,SUM,SQA,ALLNUC,PSA,LSQA),
REMOTE=(SYSLIST=*('OMVS'),SDATA,DSPNAME),END
```

- Issue command on only one system in shared FS plex, and dump of OMVS and its dataspaces will be taken on all systems in shared FS plex
- SYS1.PARMLIB(IEADMCxx) - dump Parmlib member
  - **DUMP PARMLIB=xx**

SLIP to Get Dumps on All Systems in Shared FS Sysplex

- The following sliptrap makes use of the REMOTE parameter to dump all systems in the shared FS sysplex.
- In this example, the sliptrap will dump OMVS and its dataspaces on all systems in the shared FS sysplex on the occurrence of an abend30D:

```
SLIP SET,COMP=30D,ACTION=SVCD,ID=S30D,
DSPNAME=('OMVS'.*),JL=OMVS,
SDATA=(PSA,CSA,LSQA,LPA,ALLNUC,RGN,SQA,SUM,TRT),
REMOTE=(SYSLIST=(SYSBPX.*),SDATA,JOBLIST,DSPNAME,A=SVCD),
END
```

- Note the difference in syntax with REMOTE parameter as compared to console dump

USS Reason Code SLIP: z/OS R7

- A generic sliptrap can be used to get a dump of a specific z/OS USS reason code (reason code qualifier in range 0-20FF):

```
SLIP SET,IF,A=SYNCSVCD,RANGE=(10?+8C?+F0?+1F4?),
DATA=(13R??+1B0,EQ,xxxxxxxx), DSPNAME=('OMVS'.*),
SDATA=(ALLNUC,PSA,CSA,LPA,TRT,SQA,RGN,SUM),
J=jobname,JL=OMVS,END
```

  - where 'xxxxxxxx' is the 8 digit (4 byte) reason code that is to be trapped and 'jobname' is the optional jobname associated with the error
  - use wildcard '*' for jobname of forked/spawned address spaces
    - **Cannot** be used to SLIP on reason codes from other components
- IEASLPxx in SYS1.PARMLIB can be used to store the sliptrap
  - **SET SLIP=xx**
- Documented in z/OS USS Messages and Codes

USS Reason Code SLIP: z/OS R8 - R9

- A generic sliptrap can be used to get a dump of a specific z/OS USS reason code (**including PFS reason codes**):

```
SLIP SET,IF,A=SYNCSVCD,RANGE=(10?+8C?+F0?+1F4?),
DATA=(13R??+1B0,EQ,xxxxxxxx), DSPNAME=('OMVS'.*),
SDATA=(ALLNUC,PSA,CSA,LPA,TRT,SQA,RGN,SUM),
J=jobname,JL=OMVS,END
```

  - where 'xxxxxxxx' is the 8 digit (4 byte) reason code that is to be trapped and 'jobname' is the optional jobname associated with the error
  - use wildcard '*' for jobname of forked/spawned address spaces
- IEASLPxx in SYS1.PARMLIB can be used to store the sliptrap
  - **SET SLIP=xx**
- Documented in z/OS USS Messages and Codes
- APAR **OA17911** rolls this z/OS R8 USS reason code SLIP enhancement for PFS reason codes back to z/OS R7

Collecting OMVS CTRACE Records

- The OMVS CTRACE is especially valuable in SLIP dumps
  - History of Syscalls up to the SLIP event
- Turn on OMVS CTRACE with console command
  - TRACE CT,4M,COMP=SYSOMVS
    - Buffer can be 64M at z/OS R7 and higher releases
  - R xx,OPTIONS=(ALL),END
    - JOBNAME can be specified (refers to "userid" associated with job or user)
      - For OMVS kernel trace entries, specify JOBNAME of OMVS
- Verify OMVS CTRACE status
  - D TRACE,COMP=SYSOMVS
- Turn off CTRACE after SLIP or console dump
  - TRACE CT,OFF,COMP=SYSOMVS
- Ensure fix applied for APAR **OA14799**
  - Some OMVS ctrace entries missing from IPCS dump taken shortly after dynamically increasing ctrace buffer size; may cause IPCS ctrace display to hang
- For details of sending OMVS CTRACE records to an External Writer, refer to z/OS MVS Diagnosis: Tools and Service Aids
  - See also ***Appendix E***

OMVS CTRACE at IPL Time

- The CTIBPX00 Parmlib member is used to start the OMVS CTRACE at IPL time.
  - Uncomment the 'ON' statement, the 'ALL' options statement, and set the BUFSIZE as required with the max being 4M (64M at z/OS R7 and higher releases)
    - CTIBPX00 is the default member for OMVS CTRACE startup. The BPXPRMxx member CTRACE statement may have been used to designate a different OMVS CTRACE Parmlib member, so you will need to check in BPXPRMxx before assuming the default.
- CTRACE usage is documented in z/OS MVS Diagnosis: Tools and Service Aids

IPCS Diagnostics

Determine Address Spaces Dumped

- IP CBF RTCT
  - Displays what address spaces are dumped in SVC dump
  - F ASTB

```
RTCT: 00FB7470
 ASTB

 001 000E  80    00
 002 001F  80    00
 003 0021  00    00
 004 0000  00    00
 005 0000  00    00
```

Determine Jobnames of Address Spaces

- IP SELECT ASID(x'nn',x'nn',...)
  - Displays jobnames associated with ASIDs
  - IP SELECT ASID(x'E',x'1F',x'21')

```
ASID JOBNAME  ASCBADDR  SELECTION CRITERIA
 ---- -------- --------  -----------------
 000E OMVS     00FA1800  ASID
 001F ETCRC    00FA0880  ASID
 0021 WELLIE3  00FA0580  ASID
```

- IP SELECT ALL
  - Displays ASIDs and jobnames for all active address spaces in the system at the time of the dump
    - Can search on specific jobname or ASID

Check for Latch Contention

- IP ANALYZE RESOURCE

```
RESOURCE #0009:
   NAME=SYS.BPX.A000.FSLIT.FILESYS.LSN  ASID=000D  Latch#=2
RESOURCE #0009 IS HELD BY:
  JOBNAME=OMVS      ASID=000D  TCB=008D97A0
   DATA=EXCLUSIVE  RETADDR=9FF1A8B2  REQID=000DF0001FC4E2D0
RESOURCE #0009 IS REQUIRED BY:
  JOBNAME=OMVS      ASID=000D  TCB=008D8030
   DATA=EXCLUSIVE  RETADDR=A002140E  REQID=000DF000184B5AD0
RESOURCE #0010:
   NAME=SYS.BPX.A000.FSLIT.FILESYS.LSN  ASID=000D  Latch#=31
RESOURCE #0010 IS HELD BY:
  JOBNAME=OMVS      ASID=000D  TCB=008D97A0
   DATA=EXCLUSIVE  RETADDR=9FF18E74  REQID=000DF0001FC4E2D0
RESOURCE #0010 IS REQUIRED BY:
  JOBNAME=MEGA1     ASID=01D5  TCB=008E1B60
   DATA=SHARED     RETADDR=9FFD33A6  REQID=01D501C4184B0A50
```

- Note that OMVS TCB(x'8D97A0') holds both FILESYS Latch#2 (MOUNT Latch) and FILESYS Latch#31

OMVSDATA

- Use OMVSDATA in a dump containing OMVS address space, and preferably at least SYSZBPX1 and SYSZBPX2 dataspaces
- Use PROCESS SUMMARY option to:
  - List Kernel status
  - List some startup (BPXPRMxx) options
  - List any USS processes showing parent/child relationships
- Use PROCESS DETAIL option to:
  - List processes and threads
  - Display the name of the program exec'd
  - Display file descriptor information
  - Display UIDs and GIDs
  - Display TCB address
  - Display syscall information

OMVSDATA (continued)

- Use FILE SUMMARY option to:
  - List FILESYSTYPE information, and specific information relating to mounted file systems of that FILESYSTYPE
  - For each file system, Latch# and Quiesce Latch# (if exist) are shown
  - In shared FS configuration, Owner of file system is displayed
- Use FILE DETAIL option to:
  - List all files in use

OMVSDATA SUMMARY

- IP OMVSDATA SUMMARY

```
 IPCS OUTPUT STREAM ----------------------------------- Line 0 Cols 1 78
****************************** TOP OF DATA ***********************************
  * * * * OPENMVS REPORT  * * * *

   Report(s):              PROCESS

   Level(s):               SUMMARY

   Filter(s):              NONE

Suppression-On-Protection is installed

Kernel status:             Active

Kernel address space name: OMVS

Kernel address space ID:   X'000E'

Kernel stoken:             0000003800000001

 Command ===>                                          SCROLL ===> CSR
```

OMVSDATA SUMMARY (cont)

```
IPCS OUTPUT STREAM ------------------------------------ Line 22 Cols 1 78
 Startup options

    Parmlib member:                                  BPXPRM00

    CTRACE parmlib member:                           CTIBPX00

    Maximum processes on system:                          500

    Maximum users on system:                              500

    Maximum processes per user id:                        500

    Maximum thread tasks per process:                      50

    Maximum threads per process:                        1,000

    Maximum allocated files per process:               60,000

    Maximum growth for pseudo-terminal sessions: 1,024

    Maximum growth for remote-terminal sessions: 1,024

Command ===>                                                  SCROLL ===> CSR
```

OMVSDATA SUMMARY (cont)

```
IPCS OUTPUT STREAM ----------------------------------- Line 44 Cols 1 78

 Stack Information: N/A


  ***** PROCESS SUMMARY REPORT *****

 Process      Userid     Asid     Parent       Process     Session   Status
   ID                              PID         Group ID      ID
 ------------------------------------------------------------------------
-
 00000001     IBMUSER    0013     00000000     00000001    00000001   -
 01000002     WELLIE0    001E     00000001     01000002    01000002   -
 01000003     WELLIE0    001E     01000002     01000003    01000003   -
 00000005     WELLIE0    001C     01000003     00000005    01000003   -
 00000006     WELLIE0    001D     00000005     00000005    01000003   -
 00000007     WELLIE0    0018     00000006     00000005    01000003   -

 No exceptional conditions were found by the PROCESS SUMMARY report.

Command ===>                                                SCROLL ===> CSR
```

- Note that all active processes are descendants of PID 1
- In this example, PID 7 is the youngest child process

OMVSDATA DETAIL

- IP OMVSDATA DETAIL ASID(x'18')

```
***** PROCESS DETAIL REPORT *****

 Process ID: 00000007
  Status: Active
  Maximum thread tasks allowed:   1,000
  Maximum threads allowed:          200

  Last exec() Program Name:
   /bin/sleep

  ID Data:
              Userid:  WELLIE0                    Asid:  0018
          Parent PID:  00000006      Ptrace Parent PID:  00000000
    Process Group ID:  00000005             Session ID:  01000003
           Real Uid:  0                       Real Gid:  512
       Effective Uid:  0                   Effective Gid:  512
       Saved Set Uid:  0                   Saved Set Gid:  512

  Signal Data (Process Level):

    Signal      Sa_Action      Sa_Flags          Sa_Mask
    ------      ---------      --------          -------
    SIGIOER     00000001       00000001     00000000 00000000
```

OMVSDATA DETAIL (cont)

```
Shared memory attaches: None
Semaphore Adjustments: None
Memory Map Files: None

File System Data:

  Current Working Directory (CWD) Name (at time of last chdir()):
      /

  CWDs File Serial Number: 00000000         CWDs Device Number: 00000003
  Number of open files for this process:               3  Token: 00406540

  fd=00000000  PathName at time of open():
   /dev/ttyp0000

  File Serial Number: 00000497       Device Number: 00000001
Device Major Number: 00000002 Device Minor Number: 00000000
         Open Flags: 00000423     Tokens: 005004B0 00408754 00382C10
          File Type: CHARSPEC         File Cursor: 00000000 00000000
  Descriptors sharing this open: 4  Opened: 07/11/2006 12:17:02.354031
```

OMVSDATA DETAIL (cont)

```
 Thread Data: Active Threads

Thread ID: 25477000 00000000

  Tcb Address:     008EC2E0

  In Kernel Call:  BPX1SLP

  History of system calls (latest entry last):
                                                    BPX1FCT
                                                    KRNEXIT
                                                    BPX2ITY
                                                    KRNEXIT
                                                    BPX1FST
                                                    KRNEXIT
                                                    BPX1FCT
                                                    KRNEXIT
                                                    BPX2ITY
                                                    KRNEXIT
                                                    BPX1SLP
```

OMVSDATA FILE SUMMARY

- IP OMVSDATA FILE SUMMARY

```
File System Type specific information
-------------------------------------

Type:   ZFS    (from parmlib FILESYSTYPE stmt)
Status: Active       Token: 002807D0

PathConf Data:
          Pipe_buf: 00000000    Posix_Chown_restricted? N


File System Type specific information
-------------------------------------

Type:   TFS    (from parmlib FILESYSTYPE stmt)
Status: Active       Token: 00280960

PathConf Data:
          Pipe_buf: 00000000    Posix_Chown_restricted? N
```

OMVSDATA FILE SUMMARY (cont)

```
Mounted File System specific information
----------------------------------------

Mounted File System Name: MEG1.OMVS.SYS1

Mount Point:
  Path Name (at time of mount):
  /u/mega
  File Serial Number: 0000CF00

Device Number: 000000ED    DD Name: SYS27351    Token: 00304C00
Owner: N/A                 Latch#:    68      Quiesce Latch#:    70

Number of active files for this mounted file system:              1
Number of files pending inactive:                                 0

PathConf Data:  Link_max: 00010000   Name_max: 000000FF
                Posix_No_trunc? Y    Posix_Chown_restricted? Y
```

OMVSDATA FILE DETAIL

- IP OMVSDATA FILE DETAIL

```
Files In Use:

File Serial Number:    00000000          Device Number: 40000003
  File Status:         Active                    Token: 00380000
  File Type:           SOCKET         File System Type: UDS
  Total number of opens for this file:                        1

File Serial Number:    000007E5          Device Number: 00000003
  File Status:         Active                    Token: 003800D0
  File Type:           REGFILE        File System Type: ZFS
  File Name (1-15): /etc/log
  Total number of opens for this file:                        1

File Serial Number:    00000100          Device Number: 00000014
  File Status:         Active                    Token: 00380410
  File Type:           DIR            File System Type: AUTOMNT
  File Name (1-15): websrv
  Total number of opens for this file:                        0
  Number of processes that use this file as CWD:              0
  Name of File System Mounted Here:
      SYS0.OMVS.S4S01.WEBSRV.WEBLOGS.HFS
```

Formatting OMVS CTRACE in IPCS

- IPCS 2.7.1.D (Analysis --> Traces --> Ctrace --> Display)

```
    ----------------- CTRACE DISPLAY PARAMETERS---------------------
COMMAND ===>

  System      ===>            (System name or blank)
  Component   ===> SYSOMVS    (Component name (required))
  Subnames    ===>

  GMT/LOCAL   ===> L                        (G or L, GMT is default)
  Start time  ===>                          (mm/dd/yy,hh:mm:ss.dddddd or
  Stop time   ===>                           mm/dd/yy,hh.mm.ss.dddddd)
  Limit       ===> 0          Exception ===>
  Report type ===> FULL       (SHort, SUmmary, Full, Tally)
  User exit   ===>            (Exit program name)
  Override source ===>
  Options         ===>

  To enter/verify required values, type any character
  Entry IDs ===>   Jobnames ===>   ASIDs ===>   OPTIONS ===>   SUBS ===>

  CTRACE COMP(SYSOMVS) LOCAL FULL

  ENTER = update CTRACE definition.  END/PF3 = return to previous panel.
  S = start CTRACE.  R = reset all fields.
```

OMVS CTRACE – Syscall Entry Record

- At z/OS R6 and higher releases

```
SYS1      SYSCALL   0F080001  16:10:24.476921  STANDARD SYSCALL ENTRY TRACE
    ASID..02AD      USERID....PUBLIC    STACK@....21248010
    TCB...009FD098  EUID......000003E6  SYSCALL...0000002F
   +0000  0000002F  00000000  D1C3E2C5  8C000000  | ........JCSE.... |
   +0010  0002000E  00000000  8598E5B4  212496F4  | ........eqV...o4 |
   +0020  21249768  00000000  00000000  00028C08  | ..p............. |
   +0030  00000007  00028C28  1F221348  00028C2C  | ................ |
   +0040  00028AC8  00028C30  1F1D0080  9F1BFFE0  | ...H...........\ |
```

- +18 RETURN ADDR TO SYSCALL CALLER (REG14)
- +2C ENTRY PARMLIST POINTER (REG 1)
- +30 NO OF PARMS IN PARMLIST

OMVS CTRACE – Syscall Exit Record

- All releases up through z/OS R9

```
SYS1      SYSCALL   0F080002  16:10:24.476922  STANDARD SYSCALL EXIT TRACE
   ASID..02AD      USERID....PUBLIC    STACK@....21248010
   TCB...009FD098  EUID......000003E6  SYSCALL...0000002F
  +0000  0000002F  00000000  D1C3E2C5  8C00236A  | ........JCSE.... |
  +0010  0002000B  00000000  FFFFFFFF  00000081  | ................a |
  +0020  053B006C                                | ...%             |
```

- +18 RETURN VALUE (RV)
- +1C RETURN CODE (ERRNO)
- +20 REASON CODE (ERRNOJR)

Locating the Caller of a USS Syscall in a Dump

- Find the appropriate TCB
- Find the appropriate linkage stack entry
  - **IP SUMM FORMAT ASID(x'nn')**
    - Displays address space information and linkage stack entries for TCBs in address space
- Determine whether FASTPATH Syscall or Non-FASTPATH Syscall
  - Find caller's PSW address and syscall parms
  - If LE environment, find CAA and DSA
    - Use LEDATA command to view LE traceback

Locating the Syscall Caller: Find the Appropriate TCB

- If dump captured is result of a SLIP on an errnojr, TCB of interest will be in **IP STATUS REGS** as the home TCB

```
CPU STATUS:
PSW=070C5000 9F27DE42
    (Running in AR, key 0, AMODE 31, DAT ON)
    DISABLED FOR PER
   ASID(X'000D') 1F27DE42. AREA(Subpool252Key00)+178E42 IN  EXTENDED PRIVATE
   ASID(X'000D') 1F27DE42. BPXINPVT+178E42 IN EXTENDED PRIVATE
  ASCB680 at F2F680, JOB(CTS7094F), for the home ASID
  ASXB680 at 9FDD00 and TCB680E at 9CDD40 for the home ASID
  HOME ASID: 02A8 PRIMARY ASID: 000D SECONDARY ASID: 02A8
```

- If a console dump, then correct TCB may be obtained from OMVSDATA, CTRACE, or it may already be known

Locating the Syscall Caller: Find the Appropriate Linkage Stack Entry

- IP SUMM FORMAT ASID(x'nn')
  - F 'TCB: 00nnnnnn'
  - F LINKAGE
- Syscall represented by LSE (linkage stack entry) with PC TARG of 13xx
- TARG with 13xx indicates PC into z/OS USS
  - 1300 space switch regular
  - 1301 non-space switch regular
  - 1302 space switch authorized
  - **1307 - 13nn FASTPATH SYSCALL**
- Fastpath Syscalls
  - shorter pathlength for performance reasons

Locating the Syscall Caller: Linkage Stack Entry Example #1

```
LINKAGE STACK ENTRY  01  LSED: 7F7160B0
 LSE: 7F716010
    GENERAL PURPOSE REGISTER VALUES
    00-03.... 00000036  0014CA14  09F25BFC  8A03388E
    04-07.... 09F2528C  00000000  00000000  FFFFFFFF
    08-11.... 0014C928  00000000  0000025C  09F25D84
    12-15.... 00016920  0014C950  8A033982  00001300
    ACCESS REGISTER VALUES
    00-03.... 00000000  00000000  00000000  00000000
    04-07.... 00000000  00000000  01FF0004  01FF0004
    08-11.... 00000000  00000000  00000000  00000000
    12-15.... 00000000  00000000  000257C0  00000000
    PKM...... 00C0      SASN..... 003B      EAX..... 0000
    PASN..... 003B      PSW...... 078D1400  81045576
    TARG..... 0000130B  MSTA..... 0A24AF28  0A24B6D8
    TYPE..... 05
        PC STATE ENTRY                          FASTPATH
    RFS...... 0F38      NES...... 00A8
```

Locating the Syscall Caller: Linkage Stack Entry => FASTPATH

- LSE contains registers at the time syscall was issued
- LSE GPR0 contains syscall number
- LSE GPR1 is the syscall parmlist pointer
- LSE GPR12 is CAA
- LSE GPR13 is DSA (non-XPLINK environment)
- LSE GPR 14 is caller of syscall

```
LINKAGE STACK ENTRY  01  LSED: 7F7160B0
LSE: 7F716010
GENERAL PURPOSE REGISTER VALUES
00-03....  00000036  0014CA14  09F25BFC  8A03388E
04-07....  09F2528C  00000000  00000000  FFFFFFFF
08-11....  0014C928  00000000  0000025C  09F25D84
12-15....  00016920  0014C950  8A033982  00001300
```

Locating the Syscall Caller: Linkage Stack Entry Example #2

```
LINKAGE STACK ENTRY  03  LSED: 7F3FA388
 LSE: 7F3FA268
    GENERAL PURPOSE REGISTER VALUES
    00-03.... 00000055  2551F378  25501C90  7F855D58
    04-07.... 017BDE00  824C8C50  2551BD90  2551BC94
    08-11.... 2551BC88  2551BDBC  02684300  00FC91B8
    12-15.... 02683C6C  25501B08  00001300  82682B68

    ACCESS REGISTER VALUES
    00-03.... 19A987C8  00000000  00000000  1E24EF24
    04-07.... 1B4CC100  9E24EF34  1E3DED2C  99BB980A
    08-11.... 1E24ED28  99BB982A  00000000  00000000
    12-15.... 00000000  00000000  00000000  00000072
    PKM...... 00C0      SASN..... 0030      EAX...... 0000
    PASN..... 0030      PSW...... 07643000  075B1722
    TARG..... 00001300  MSTA..... 1D052010  1D052FB8
    TYPE..... 0D
     PC STATE ENTRY
    RFS...... 0C58      NES...... 0128
```

Non-FASTPATH

Locating the Syscall Caller: Linkage Stack Entry => Non-FASTPATH

- LSE GPR0 contains syscall number
- LSE GPR1 is the syscall parmlist pointer
- LSE GPR2 contains pointer to USTA

```
LINKAGE STACK ENTRY  03  LSED: 7F3FA388
LSE: 7F3FA268
GENERAL PURPOSE REGISTER VALUES
00-03....  00000055  2551F378  25501C90  7F855D58
04-07....  017BDE00  824C8C50  2551BD90  2551BC94
08-11....  2551BC88  2551BDBC  02684300  00FC91B8
12-15....  02683C6C  25501B08  00001300  82682B68
```

- Format USTA as follows:
  - **IP CBF *GPR2_VALUE* STR(BPXZUSTA) ASID(x'nn')**
- USTA is control block that contains registers at time of the syscall
  - Always verify that GPR0 contains syscall number

Locating the Syscall Caller: Linkage Stack Entry => Non-FASTPATH (cont)

- IP CBF 25501C90 STR(BPXZUSTA) ASID(x'30')

```
USTA: 25501C90
 +0000   ID              E4E2E3C1
 +0004   SP              83
 +0005   LEN             0000F8
 +0008   FLAGS           00000000
 +0010   PSWBYT03        078D3400   Caller of
 +0014   PSWIC           86025D44   Syscall
 +0018   GR0             00000055
 +0048   GR12            2551A4B8   CAA
 +004C   GR13            255202E0   DSA (non-XPLINK)
 +0050   GR14            86025D44
 +0054   GR15            826B0140
```

Use LEDATA to View LE Traceback

- IP VERBX LEDATA 'TCB(007DE1B8) CAA(00016920) DSA(0014C950) ASID(3B) ALL‘
  - Note: substitute appropriate values for TCB, CAA, DSA and ASID

```
Traceback:
Entry           E Addr     E  Offset   Statement  Load Mod  Service  Status
write           0A033840   -09C9441A                                 Call
wp124           0A1D8530   +0000052E                                 Call
fwrite          0A13E978   +0000026C                                 Call
filebuf::overflow(int)
                0A336530   +000004DC                                 Call
ostream::flush()
                0A3753C8   +0000067A                                 Call
endl(ostream&)
                0A353C28   +00000472                                 Call
ostream::operator<<(ostream&(*)(ostream&))
                09F0D660   +00000052                                 Call
CEEVSSFR        0000A8C8   +00000012                                 Call
main            09F09D88   +0000008A                                 Call
CEEVSSFR        0000A8C8   +00000012                                 Call
EDCZMINV        0A1C6B3E   +000000B4                                 Call
CEEBBEXT        0000E328   +0000013C                                 Call
```

Problem Scenarios

Hard Failures

Problem #1: Terminating Signal

- Message BPXP010I indicates source of signal and receiver of signal
  - Thread ID in hex
  - Process ID in decimal.
  - Find decimal PID using: D OMVS,A=ALL
    - PID 67108975 = x'0400006F' (for OMVSDATA in dump)
    - PID 16777241 = x'01000019' (for OMVSDATA in dump)

BPXP010I THREAD 352E800000000000, IN PROCESS 67108975, WAS TERMINATED BY SIGNAL SIGKILL, SENT FROM THREAD 000000000000000, IN PROCESS 16777241, UID 0.

IEF450I G113821 STEP1 - ABEND=SEC6 U0000 REASON=0000FF09

- Abend EC6 REASON=0000FF09
  - FF09 = SIGKILL

Problem #2: OMVS Initialization Error

- Customer received following error during IPL of SY1:
  - **BPXI027I THE ETCINIT JOB ENDED IN ERROR, EXIT STATUS 00000900**
  - Per USS Messages and Codes: Exit Status Codes for /usr/sbin/init

09 No stdout
/usr/sbin/init could not open /etc/log for standard output. Contact your system programmer. The system continues.
.
Note: /usr/sbin/init attempts to erase and re-create /etc/log each time it is run.

  - Possible Causes
    - Full file system, improper authorization, or read-only /etc file system

Problem #2: OMVS Initialization Error D OMVS,F

- Use D OMVS,F to display /etc file system for SY1

```
SY1  D OMVS,F
SY1  BPXO045I 15.19.06 DISPLAY OMVS 231
OMVS     000D ACTIVE          OMVS=(MN)
TYPENAME   DEVICE ----------STATUS---------- MODE
ZFS             4 ACTIVE                    READ
  NAME=ZOS16.SY1.ETC.ZFS
  PATH=/SY1/etc
  OWNER=SY1      AUTOMOVE=Y CLIENT=N
```

- Note that SY1 /etc file system 'ZOS16.SY1.ETC.ZFS' is mounted as read-only

Problem #3: Process Limit Exceeded

- Websphere began to fail with:
  - **IMW0501E Accept soft error:**

    Errno: 124 Errno2: 050B0146 Error: EDC5124I Too many open files.
- RC 124 - EMFILE - Too many files are open for this process.
- BPXMTEXT 050B0146

BPXFRNXT 09/17/01
JROpenMax: The maximum number of open files for this process was reached

Action: Close any file descriptors that are no longer needed.

Problem #3: Process Limit Exceeded
D OMVS,L,PID=

- Use D OMVS,L,PID=xxx to display process limits

```
D OMVS,L,PID=129
BPXO051I 07.13.02 DISPLAY OMVS 179
OMVS     000D ACTIVE          OMVS=(6E)
USER     JOBNAME  ASID        PID       PPID STATE   START     CT_SECS
IBMUSER  PSTORE   008C        129          1 MRI--- 19.41.51        .83
  LATCHWAITPID=          0 CMD=BBOCTL
PROCESS LIMITS:         LIMMSG=NONE
                  CURRENT   HIGHWATER    PROCESS
                    USAGE      USAGE      LIMIT
MAXFILEPROC           512        512        512
MAXFILESIZE           ---        ---    NOLIMIT
MAXPROCUSER             1          4    NOLIMIT
MAXQUEUEDSIGS           0          1       1000
MAXTHREADS              0          0        200
MAXTHREADTASKS          0          0       1000
IPCSHMNSEGS             0          0        500
MAXCORESIZE           ---        ---    4194304
```

- Consider using SETOMVS to raise MAXFILEPROC value

Problem #4: System Limit Exceeded

- Attempts to start new OMVS processes failed as follows:
  - **BPXP012I FORK SYSCALL TERMINATED DURING CHILD PROCESSING WITH RETURN CODE 0000009C, REASON CODE 0B0F0028.**
- RC x'9C' - EMVSINITIAL - Process Initialization Error
- BPXMTEXT 0B0F0028

BPXPRTB1 07/15/02
JRMaxProc: The maximum number of processes was exceeded

Action: Retry after some processes have ended, or change the maximum number of processes allowed.

Problem #4: System Limit Exceeded D OMVS,L

- Use D OMVS,L to display system limits

```
D OMVS,L
BPXO051I 06.22.07 DISPLAY OMVS 171
OMVS     000D ACTIVE            OMVS=(6E)
SYSTEM WIDE LIMITS:           LIMMSG=NONE
                    CURRENT  HIGHWATER     SYSTEM
                      USAGE       USAGE      LIMIT
MAXPROCSYS              256         256        256
MAXUIDS                   0           0        200
MAXPTYS                   0           0        256
MAXMMAPAREA               0           0        256
MAXSHAREPAGES             0          10       4096
IPCMSGNIDS                0           0        500
IPCSEMNIDS                0           0        500
IPCSHMNIDS                0           0        500
IPCSHMSPAGES              0           0     262144
IPCMSGQBYTES            ---           0 2147483647
IPCMSGQMNUM             ---           0      10000
IPCSHMMPAGES            ---           0      25600
```

MAXPROCSYS limit value can be raised via the SETOMVS command.

Problem #4: System Limit Exceeded
D OMVS,A=ALL

- One reason for exceeding MAXPROCSYS is due to a buildup of zombies
- D OMVS,A=ALL

```
d omvs,a=all
BPXO040I 17.25.48 DISPLAY OMVS 202
OMVS     000D ACTIVE          OMVS=(6F)
USER     JOBNAME  ASID        PID       PPID STATE   START     CT_SECS
WASPISRV IMWEBJBP 0021         196637       1 MR---- 10.08.13      .629
  LATCHWAITPID=          0 CMD=IMWHTTPD
WSPUBLIC           0000 0050528627     196637 1Z---- 14.02.48      .001
WSPUBLIC           0000 0084083060     196637 1Z---- 14.07.10      .001
WSPUBLIC           0000 0016974201     196637 1Z---- 11.00.57      .001
WSPUBLIC           0000 0033751418     196637 1Z---- 14.07.00      .001
WSPUBLIC           0000 0000196987     196637 1Z---- 11.35.43      .001
WSPUBLIC           0000 0067305854     196637 1Z---- 14.18.08      .001
WSPUBLIC           0000 0033751428     196637 1Z---- 11.05.13      .001
WSPUBLIC           0000 0016974213     196637 1Z---- 13.54.18      .001
```

- Note that there is a buildup of zombies, all with a parent PID of 196637
- This should be pursued by Webserver support

Problem Scenarios

Hangs

Problem #5: User Hangs When FTPing to Host (Non-Shared FS)

- Customer collected console dump of user FTP session (IBMUSER), along with OMVS and its dataspaces
- **IP CBF RTCT; IP SELECT ASID(x'1F')**
  - IBMUSER is asid(x'1F')
  - OMVS is asid(x'E')
- **IP OMVSDATA SUMMARY**
  - IBMUSER is a child of FTPD daemon
- **IP OMVSDATA DETAIL ASID(x'1F')**
  - Last exec() Program Name: /usr/sbin/ftpdns
  - Thread ID:255AD800 00000002
    - TCB Address: 008EC020
    - In Kernel Call: BPX1STA

Problem #5: User Hangs When FTPing to Host BPX1STA – Assembler Callable Services

- Function
  - The stat callable service obtains status information about a specified file. You specify the file by its pathname.
- BPX1STA Parameters - USS Assembler Callable Services
  - Pathname_length
  - Pathname
  - Status_area_length
  - Status_area
  - Return_value
  - Return_code
  - Reason_code

Problem #5: User Hangs When FTPing to Host BPX1STA Linkage Stack / Parameters

- **IP SUMM FORMAT ASID(x'1F')**
  - FIND 'TCB: 008EC020'
  - FIND LINKAGE
    - Register 0 = Syscall Number: x'2F' STAT
    - Register 1 = Parmlist: 00164F68
- **IP L 00164F68 ASID(x'1F') LEN(28)**
  - 00164F68. 00164F8C 00122C08
- **IP L 00164F8C ASID(x'1F') LEN(4)**
  - 00164F8C. 00000008
  - Pathname length = 8
- **IP L 00122C08 ASID(x'1F') LEN(8)**
  - 00122C08. 6185A383 61939687
  - Pathname = /etc/log

Problem #5: User Hangs When FTPing to Host OMVSDATA FILE SUMMARY

- In what filesystem is /etc/log ?
  - After OS/390 R9, /etc is a symbolic link
    - $SYSNAME/etc
      - $SYSNAME = /SYSTEM on non-shared FS systems
- **IP OMVSDATA FILE SUMMARY**

```
Mounted File System specific information
----------------------------------------
Mounted File System Name: POSIX.HFS.ETC
Mount Point:
  Path Name (at time of mount):
  /SYSTEM/etc
  File Serial Number: 00000016
Device Number: 00000004    DD Name: SYS00002    Token: 00300600
Owner: N/A                 Latch#:    14      Quiesce Latch#:    18
Number of active files for this mounted file system:              4
Number of files pending inactive:                                 0
PathConf Data:  Link_max: 00010000   Name_max: 000000FF
                Posix_No_trunc? Y    Posix_Chown_restricted? Y
This file system has been quiesced
```

Problem #5: User Hangs When FTPing to Host ANALYZE RESOURCE

- As mentioned earlier, a quiesced HFS file system can cause FILESYS latch contention
- **IP ANALYZE RESOURCE**

```
CONTENTION REPORT BY RESOURCE NAME
 RESOURCE #0001:
    NAME=SYS.BPX.A000.FSLIT.FILESYS.LSN  ASID=000E  Latch#=18
 RESOURCE #0001 IS HELD BY:
   JOBNAME=MEGA5     ASID=0043  TCB=008EC1A0
    DATA=EXCLUSIVE  RETADDR=A5115432  REQID=C2D7E7C6E2D8E2C5

 RESOURCE #0001 IS REQUIRED BY:
   JOBNAME=IBMUSER   ASID=001F  TCB=008EC020
    DATA=SHARED     RETADDR=A52189E6  REQID=001F000D24D21080
   JOBNAME=MEGA5     ASID=0043  TCB=008EC1A0
    DATA=SHARED     RETADDR=A52189E6  REQID=0043000A24D20EC0
```

Problem #5: User Hangs When FTPing to Host View File System in ISHELL Mount Table

- Use ISHELL to view the Mount Table

```
File  Directory  Special_file  Tools  File_systems  Options  Setup  Help
 ------------------------------------ .----------------------. ----------
                      UNIX System Serv | 1  1. Mount table...  |
                                       |    2. New...          |
 Enter a pathname and do one of these: |    3. Mount(O)...     |
                                       '----------------------'

     - Press Enter.
     - Select an action bar choice.
     - Specify an action code or command on the command line.

 Return to this panel to work with a different pathname.
                                                          More:     +
    /
    ___________________________________________________________________
    ___________________________________________________________________
    ___________________________________________________________________
```

Problem #5: User Hangs When FTPing to Host Fixing the Problem With ISHELL

- Reset quiesce on mounted file system

```
Work with Mounted File Systems

 Select one or more file systems with / or action codes.
  U=Unmount   A=Attributes   M=Modify   R=Reset unmount or quiesce
    File system name                Status              Row 1 of 5
 R POSIX.HFS.ETC                    Quiesced by MEGA5 (50331660)
 _ POSIX.HFS.FS                     Available
 _ POSIX.HFS.MAN                    Available
 _ POSIX.HFS.NLS                    Available
 _ POSIX.USR.LPP                    Available
```

Problem #6: Shared FS Hang

- Customer reported message BPXF076I when trying to IPL system in shared FS sysplex.
  - **BPXF076I FILE SYSTEM INITIALIZATION IS DELAYED DUE TO CONFLICTING ACTIVITY ON ANOTHER SYSTEM.**
- Customer was running with 24 z/OS R7 systems in a shared FS sysplex. At the time of the problem, 22 systems were active (SYS4 and SYS2 were down).
- Customer attempted to bring SYS2 back up, OMVS initialization hung, and they got message BPXF076I.

Problem #6: Shared FS Hang (cont)

- The 'F BPXOINIT,FILESYS=D,GLOBAL' command showed the following:

```
SYSTEMS PERFORMING LOCAL FILE SYSTEM RECOVERY
  (Since 2007/06/09 01.00.00)
      SYS3
SYSTEMS PERFORMING FILE SYSTEM TAKEOVER RECOVERY
  (Since 2007/06/09 01.00.00)
      SYS3
SYSTEMS PERFORMING UNMOUNT
  (Since 2007/06/08 14.30.00)
      SYS1
```

- Note that UNMOUNT processing on SYS1 is the earliest shared FS activity, hung up since the day before

Problem #6: Shared FS Hang (cont)

- SYS2 got message BPXF076I during its IPL into the shared FS sysplex because it couldn't complete its Initialization processing until both the Unmount processing on SYS1 and the Dead System Recovery processing on SYS3 completed, since they are conflicting activity.
- The 'D GRS,C' display (routed to all systems in the shared FS plex) showed contention with the MOUNT Latch on systems SYS3, SYSA, and SYS6.
- In an attempt to further isolate the problem and determine why the Unmount processing on SYS1 was hung up, the customer issued F BPXOINIT,FILESYS=FIX

Problem #6: Shared FS Hang (cont)

- The following message was issued on SYS1 as part of FIX processing:

```
BPXF049I UNMOUNT PROCESSING FOR FILE SYSTEM
POSIX.ETC.HFS REQUIRES RESPONSES FROM THE
FOLLOWING SYSTEMS:  SYS3
```

- The following message was issued on SYS3 as part of FIX processing:

```
BPXF042I POSSIBLE CONTENTION FOR THE FILE SYSTEM
MOUNT LATCH EXISTS ON SYSTEM SYS3,
LATCH NUMBER 2
```

- Taking into account the GLOBAL and FIX output, SYS3 is the problem system – IPL may be required

Reference Information

USS Programming References

- C/C++ Run-time Library Reference
- z/OS Using REXX and z/OS UNIX System Services
- z/OS USS Programming: Assembler Callable Services Reference
  - BPX1xxx, BPX2xxx, BPX4xxx
- z/OS USS Filesystem Interface Reference
  - Virtual Filesystem operations (vfs_xxx and v_xxx)
  - Vnode operations (vn_xxx)
  - Operating System Interface (OSI)

Other USS References

- Manuals
  - z/OS USS Command Reference
  - z/OS USS Messages and Codes
  - z/OS USS Planning
  - z/OS UNIX System Services User's Guide
  - z/OS MVS System Commands
  - z/OS MVS Diagnosis: Reference
  - z/OS MVS Diagnosis: Tools and Service Aids
  - z/OS MVS Initialization and Tuning Reference
- z/OS USS External Website
  - http://www-1.ibm.com/servers/eserver/zseries/zos/unix/

Related SHARE Sessions and PK100/PK300

- 8208 Diagnosing Application Problems Under LE
  - 9:30 AM Wednesday
- 8219 Introduction to IPCS for Application Programmers
  - 8:00 AM Wednesday
- For additional training in z/OS diagnosis, visit URL: www.ibm.com/training/us
  - Search on z/OS Diagnostic Workshop course #**PK100**
  - Also course #**PK300** Diagnostic Workshop for z/OS Language Environment Applications

Appendices

Appendix A: Syscall Numbers

Syscall#	Name	BPX1	Syscall#	Name	BPX1	Syscall#	Name	BPX1
1	resource	RMG	21	mkdir	MKD	41	getgid	GID
2	isatty	ITY	22	fchaudit	FCA	42	getlogin	GLG
3	ttyname	TYN	23	mknod	MKN	43	getpgrp	GPG
4	tcflush	TFH	24	mount	MNT	44	getpid	GPI
5	tcdrain	TDR	25	opendir	OPD	45	getpwnam	GPN
6	tcflow	TFW	26	open	OPN	46	getppid	GPP
7	tcgetattr	TGA	27	pathconf	PCF	47	getpwuid	GPU
8	tcgetpgrp	TGP	28	pipe	PIP	48	getugrps	GUG
9	tcsetattr	TSA	29	readdir	RDD	49	getuid	GUI
A	tcsendbreak	TSB	2A	readlink	RDL	4A	setdubdefault	SDD
B	tcsetpgrp	TSP	2B	read	RED	*4B	dub	IN1
C	access	ACC	2C	rename	REN	4C	kill	KIL
D	chdir	CHD	2D	rewinddir	RWD	4D	mvssigsetup	MSS
E	chmod	CHM	2E	rmdir	RMD	4E	pause	PAS
F	chown	CHO	2F	stat	STA	4F	ptrace	PTR
10	closedir	CLD	30	symlink	SYM	50	sigaction	SIA
11	close	CLO	31	extlink_np	EXT	51	setgid	SGI
12	w_getmntent	GMN	32	umask	UMK	*52	ptrace2	PT2
13	w_statfs	STF	33	umount	UMT	53	mvsunsigsetup	MSD
14	chaudit	CHA	34	unlink	UNL	54	sigpending	SIP
15	fchmod	FCM	35	utime	UTI	55	sleep	SLP
16	fchown	FCO	36	write	WRT	56	setpgid	SPG
17	fcntl	FCT	37	alarm	ALR	57	sigprocmask	SPM
18	fpathconf	FPC	38	exec	EXC	58	setsid	SSI
19	fstat	FST	39	_exit	EXI	59	sigsuspend	SSU
1A	fsync	FSY	3A	execmvs	EXM	5A	setuid	SUI
1B	ftruncate	FTR	3B	fork	FRK	5B	sysconf	SYC
1C	getcwd	GCW	3C	getegid	GEG	5C	times	TIM
1D	ioctl	IOC	3D	geteuid	GEG	5D	uname	UNA
1E	link	LNK	3E	getgrgid	GEU	5E	wait	WAT
1F	lseek	LSK	3F	getgrnam	GGN	*5F	operdispmvs	ODM
20	lstat	LST	40	getgroups	GGR	60	quiesce	QSE

Syscall#	Name	BPX1	Syscall#	Name	BPX1	Syscall#	Name	BPX1
61	unquiesce	UQS	81	gethostid	HST	A1	v_remove	VRM
62	MvsIptAffinity	IPT	82	getpeername	GNM	A2	v_rename	VRN
63	-	-	83	getsockopt	OPT	A3	v_fstatfs	VSF
*64	exec2	EX2	84	listen	LSN	A4	v_lockctl	VLO
65	mvsprocclp	MPC	85	readv	RDV	A5	attach_exec	ATX
66	pthread_quiesc	PTQ	86	recv	RCV	A6	attach_execmvs	ATM
67	sigputback	SPB	87	recvfrom	RFM	*A7	Xmem_kernpost	XKP
68	seteuid	SEU	88	recvmsg	RMS	*A8	Select_post	CSP
69	setegid	SEG	89	select	SEL	A9	MVSpauseInit	MPI
6A	getpsent	GPS	8A	send	SND	AA	Set_limits	STL
6B	pthread_create	PTC	8B	sendmsg	SMS	AB	MVSpause	MP
6C	pthread_xandg	PTX	8C	sendto	STO	AC	msgctl	QCT
6D	pthread_join	PTJ	8D	setpeer	SPR	AD	msgget	QGT
6E	pthread_detach	PTD	8E	shutdown	SHT	AE	msgrcv	QRC
6F	pthread_cancel	PTB	8F	socket_pair	SOC	AF	msgsnd	QSN
70	pthread_self	PTS	90	writev	WRV	B0	semctl	SCT
*71	pthread_cac	ATT	91	v_reg	VRG	B1	semget	SGT
72	pthread_setintr	PSI	92	v_rpn	VRP	B2	semop	SOP
73	pthread_kill	PTK	93	-	-	B3	shmat	MAT
74	sigwait	SWT	94	v_get	VGT	B4	shmctl	MCT
75	Pth_Setintrtype	PST	95	v_rel	VRL	B5	shmdt	MDT
76	Pth_testintr	PTI	96	v_lookup	VLK	B6	shmget	MGT
77	Cond_cancel	CCA	97	v_rdwr	VRW	B7	w_getipc	GET
78	Cond_post	CPO	98	v_readdir	VRD	B8	setpriority	SPY
79	Cond_setup	CSE	99	v_readlink	VRA	B9	getpriority	GPY
7A	Cond_timed_wait	CTW	9A	v_create	VCR	BA	nice	NIC
7B	Cond_wait	CWA	9B	v_mkdir	VMD	BB	getitimer	GTR
7C	chattr	CHR	9C	v_symlink	VSY	BC	setitimer	STR
7D	fchattr	FCR	9D	v_getattr	VGA	BD	spawn	SPN
7E	accept	ACP	9E	v_settatr	VSA	BE	chpriority	CHP
7F	bind	BND	9F	v_link	VLN	BF	pfsctl	PCT
80	connect	CON	A0	v_rmdir	VRE	C0	getgrent	GGE

Syscall#	Name	BPX1	Syscall#	Name	BPX1	Syscall#	Name	BPX1
C1	getpwent	GPE	E1	tcsetcp	TSC	101	takesocket	TAK
C2	setgrent	SGE	E2	tcsettables	TST	102	__smf_record	SMF
C3	setpwent	SPE	E3	tcgetsid	TGS	103	v_pathconf	VPC
C4	password	PWD	E4	grantpt	GPT	104	__security	SEC
C5	setgroups	SGR	E5	unlockpt	UPT	105	__wlm	WLM
C6	mmap	MMP	E6	ttyname2	2TYN	106	openstat	2OPN
C7	mprotect	MPR	E7	isatty2	2ITY	107	__getthent	GTH
C8	msync	MSY	E8	poll	POL	108	accept_and_recv	ANR
C9	munmap	MUN	E9	getwd	GWD	109	send_file PC-only	SF
*CA	vnode_tracker	VRT	EA	read_extlink	RDX	10A	__shm_lock	SLK
CB	setrlimit	SRL	EB	v_access	VAC	10B	__pid_affinity	PAF
CC	getrlimit	GRL	EC	querydub	QDB	10C	Set_Timer_Event	STE
CD	getrusage	GRU	*ED	cms_init (VM)	CTE	10D	srx_np	SRX
CE	truncate	TRU	*EE	vm_pathdef (VM)	GPH	10E	__Sigactionset	SA2
CF	lchown	LCO	EF	oe_env_np	ENV	10F	aio_suspend	ASP
*D0	kern_limit	KLM	F0	pthread_security_np	TLS	*110	createsockid	CSI
D1	waitid/wait3	WTE	F1	convert_id_np	CID	111	sigtimedwait	STW
D2	StatVfs	STV	F2	auth_check_rsrc_np	ACK	112	__osenv	OSE
D3	FstatVfs	FTV	F3	recvmsg2	2RMS	113	sigqueue	SGQ
D4	fchdir	FCD	F4	sendmsg2	2SMS	114	Pread and Pwrite	RW
D5	readdir2	RD2	F5	w_pioctl	PIO	115	__smc	SMC
D6	getpgid	GEP	F6	asyncio	AIO	116	-	-
D7	getsid	GES	*F7	asyncio2	2AIO	117	-	-
D8	sync	SYN	F8	-	-	118	sw_signaldelv	DSD
D9	chroot	CRT	*F9	kernel_med	MED	119	__mount	2MNT
DA	v_export	VEX	FA	server_init	SIN	11A	__cpl	CPL
DB	loadhfs	LOD	FB	server_pwu	SPW	11B	__map_init	MMI
DC	realpath	RPH	FC	console_np	CCS	11C	__map_service	MMS
DD	deletehfs	DEL	FD	pthread_tag_np	PTT	11D	server_thread_query	STQ
DE	setreuid	SRU	*FE	kernel_int	KIN	11E	MvsThreadAffinity	TAF
DF	setregid	SRG	FF	getclientid	GCL	11F	Pthd_quiesce_&_get_np	PQG
E0	tcgetcp	TGC	100	givesocket	GIV	120	gethostbyname	GHN

Syscall #	Name	BPX1
121	gethostbyaddr	GHA
122	GetAddrInfo	GAI
123	FreeAddrInfo	FAI
124	GetNameInfo	GNI
125	__poe	POE
126	Lchattr	LCR

Note: “*" = internal only syscalls

Some stub names are BPX2xxx (i.e. 2AIO = BPX2AIO)

Appendix B: Signal Numbers

- Terminating Signals

SIGHUP	1	Hangup detected on controlling terminal
SIGINT	2	Interactive attention, attention key pressed
SIGABRT	3	Abnormal termination, raised by abort() function
SIGILL	4	Detection of an incorrect hardware instruction
SIGPOLL	5	Pollable event
SIGURG	6	High bandwidth data is available at a socket
SIGFPE	8	Erroneous arithmetic operation, i.e.. divide by zero
SIGKILL	9	Termination (cannot be caught or ignored)
SIGBUS	10	Bus error
SIGSEGV	11	Detection of an incorrect memory reference
SIGSYS	12	Bad system call
SIGPIPE	13	Write on a pipe with no readers
SIGALRM	14	Timer expired
SIGTERM	15	Termination (application)

Terminating Signals (cont)

SIGUSR1	16	Reserved as application defined signal 1
SIGUSR2	17	Reserved as application defined signal 2
SIGABEND	18	Raised by the abend() function
SIGQUIT	24	Interactive termination
SIGTRAP	26	Communication signal used by the ptrace call
SIGXCPU	29	CPU time limit exceeded
SIGXFSZ	30	File size limit exceeded
SIGVTALRM	31	Virtual timer expired
SIGPROF	32	Profiling timer expired
SIGDANGER	33	Termination

Job Control Signals

SIGSTOP	7	Stop (cannot be caught or ignored)
SIGCONT	19	Continue if stopped
SIGCHLD	20	Child process terminated, stopped, or continued
SIGTIN	21	Read from a control terminal attempted by a member of a background process group
SIGTOU	22	Write to a control terminal attempted by a member of a background process group
SIGTSTP	25	Interactive stop
SIGTHSTOP	34	Thread stop (cannot be caught or blocked or ignored)
SIGTHCONT	35	Thread continue (cannot be caught or blocked or ignored)

Miscellaneous Signals

SIGIO	23	Completion of input or output
SIGIOER	27	Input or output error
SIGWINCH	28	Change size of window
SIGDCE	38	Reserved for exclusive use by DCE
SIGDUMP	39	Take a SYSMDUMP

Appendix C

F BPXOINIT Functions

F BPXOINIT Termination Functions

- Documented in z/OS MVS System Commands: Controlling z/OS UNIX System Services (z/OS UNIX)
- F BPXOINIT,TERM=pid.tid
  - Initiates a SIGTERM signal to the specified process id or optionally a thread id in a specified process
- F BPXOINIT,FORCE=pid.tid
  - Same as TERM= but a SIGKILL is issued instead
- F BPXOINIT,SUPERKILL=pid
  - New with z/OS R6
  - Terminates the entire process and any sub-processes within the address space
  - Only use as last resort following unsuccessful attempts to terminate the process using F BPXOINIT,TERM= and FORCE= commands
  - Target thread is abended (422-0109)

F BPXOINIT Shutdown Functions

- F BPXOINIT,SHUTDOWN=FORKINIT
  - Shuts down BPXAS fork initiators to avoid blocking P JES2 completion
- F BPXOINIT,SHUTDOWN=FORKS
  - Attempts to shutdown all running and future WLM fork initiators. Restarted with F BPXOINIT,RESTART=FORKS
- F BPXOINIT,SHUTDOWN=FILESYS
  - Non-Shared FS:
    - Synchronizes data to file systems and unmounts file systems
  - Shared FS:
    - Synchronizes data to file systems and possibly unmounts or moves ownership of the file systems owned by this system
- F BPXOINIT,SHUTDOWN=FILEOWNER
  - Shared FS only
  - Same as SHUTDOWN=FILESYS, but this system will also be disabled as a future file system owner via move or recovery operations until OMVS is recycled.

Appendix D

F OMVS Functions

OMVS Shutdown/Restart

- At z/OS R3 and higher releases, OMVS can be recycled
- F OMVS,SHUTDOWN
  - Recommended procedure for shutting down z/OS USS
  - Eliminates the need to issue the following as part of z/OS USS shutdown:
    - F BPXOINIT,SHUTDOWN=FORKINIT
    - F BPXOINIT,SHUTDOWN=FILESYS (FILEOWNER)
  - **Cannot** be used to resolve latch contention and other severe system outages
    - Refer to F BPXOINIT,RECOVER=LATCHES for resolving latch contention
  - **Cannot** be used to apply z/OS USS maintenance
    - At z/OS R7, can use 'F OMVS,ACTIVATE=SERVICE' for most USS maintenance
- F OMVS,RESTART
  - Restart z/OS USS environment

Dynamic Service Activation

- At z/OS R7, support was added to allow customers to activate most USS maintenance on running system
- F OMVS,ACTIVATE=SERVICE
  - Activate new maintenance on running system
    - ►PTFs with ++HOLD REASON(DYNACT)
  - Activations remain in effect across 'F OMVS,SHUTDOWN' and can be done during 'F OMVS,SHUTDOWN'
  - **NOTE:** Even if fix dynamically activated, need to install fix in proper library in order for fix to be available for next IPL
- F OMVS,DEACTIVATE=SERVICE
  - Backs off last set of service items activated dynamically
- D OMVS,ACTIVATE=SERVICE
  - Provides list of all sets of service items currently activated dynamically

Appendix E

OMVS CTRACE to External Writer

OMVS CTRACE to External Writer

- 1. Start an external writer for OMVS:
  - TRACE CT,WTRSTART=XWOMVS
- 2. Turn on the OMVS CTRACE:
  - TRACE CT,ON,COMP=SYSOMVS
  - reply: R xx,WTR=XWOMVS
  - reply: R xx,OPTIONS=(ALL),END
    - JOBNAME can be specified (refers to "userid" associated with job or user)
- 3. Stop the OMVS CTRACE by issuing:
  - TRACE CT,OFF,COMP=SYSOMVS
- 4. Stop the external writer by issuing:
  - TRACE CT,WTRSTOP=XWOMVS

OMVS CTRACE to External Writer (cont)

- Example Cataloged Procedure for an external writer procedure
  - The procedure places trace data on two DASD data sets
  - The procedure is placed in member XWOMVS of SYS1.PROCLIB
  - Update the DSNAME and VOLSER as required for your installation
  - External writer for OMVS CTRACE should have a dispatching priority at least equal to OMVS

```
//XWOMVS  PROC
//IEFPROC  EXEC PGM=ITTTRCWR,REGION=32M
//TRCOUT01 DD  DSNAME=SYS1.CTRACE1,VOL=SER=TRACE1,UNIT=DASD,
//        SPACE=(CYL,10),DISP=(NEW,KEEP),DSORG=PS
//TRCOUT02 DD  DSNAME=SYS1.CTRACE2,VOL=SER=TRACE2,UNIT=DASD,
//        SPACE=(CYL,10),DISP=(NEW,KEEP),DSORG=PS
```