What's New in Unix System Services?

Ann T. Totten

atotten@us.ibm.com

Monday August 22, 2005

Session 2919

Trademarks

- **The following are trademarks of the IBM Corporation in the United States or other countries or both:**
  - IBM
  - Language Environment
  - z/OS

- **UNIX** is a registered trademark of The Open Group in the United States and other countries

Agenda

- Shells and Utilities
- File Systems
- Kernel
- Health Checks

Shells and Utilities

- OpenSSH Version Update
- Pax enhancements
- Dbx enhancements

Shells and Utilities (continued)

- OpenSSH Version Update
  - Problem:
    - Current OpenSSH was ported from OpenSSH/BSD-3.5p1 so it does not have the enhancements for versions after OpenSSH/BSD-3.5p1. Many of these enhancements were security related.
  - Solution:
    - Port OpenSSH/BSD-3.8.1p1 to z/OS and merge it with the current OpenSSH.
    - For more details:
      - *See session 2919:* ***Securing Your z/OS UNIX Network Access with OpenSSH***
        On Monday 3:00 PM

Shells and Utilities (continued)

- Pax enhancements

  In this release zFS will be the preferred file system. Pax, used in copy mode, has been identified as the tool of choice for these data migrations from HFS to zFS. However, changes needed to be made to pax to do this.

  - Problem 1:
    - Pax currently causes sparse files to be expanded. Sparse files are a way of preserving disk space when storing files which contain large sections of data composed only of zeros. When a file is sparse these large sections containing zeros will not be stored on disk but when the file is read the file system will return zeros for those sections.
  - Solution 1:
    - Pax will write target files as sparse files when it is working in copy mode. There will be an option to turn this feature off.

Shells and Utilities (continued)

- Pax enhancements (continued)
  - Problem 2:
    - Currently when pax encouters an error when reading a source file while in copy mode pax will print an error message and exit.
  - Solution 2:
    - There is now an option flag to cause pax to continue processing when a read error occurs after pax has printed an error message to stderr. A non-zero value will be returned when pax exits.

Shells and Utilities (continued)

- Pax enhancements (continued)
  - Problem 3:
    - The current pax -X option causes pax to write only those files that are on the same device as the parent directory. Invoking this option causes active mountpoints to be ignored when pax is in copy mode.
  - Solution 3:
    - A new pax option will cause an empty directory to be created on the target for these mountpoints. This assists migrations if the source contains active mountpoints.

Shells and Utilities (continued)

- dbx enhancements
  - Problem:
    - When the current release of dbx encounters a context with multiple views, it will warn the user and remove all views other than the “first” one.
  - Solution:
    - When a context with multiple views is detected, dbx will only remove views if necessary. In some cases, only the views that have counterparts in the event list will not be removed. In other cases, all the views will be left intact. If there is a scenario where only one view **must** remain, and there is not enough contextual information to allow that to happen, the user will be prompted with a menu of valid contexts.

Shells and Utilities (continued)

- dbx enhancements (continued)
  - Problem:
    - dbx does not properly handle all C++ typecast expressions. In the most common case, dbx will produce an incorrect type check error message.
  - Solution:
    - The dbx scanner was changed to determine the type information for a symbol and pass that information along to the parser/traverser/evaluator.

Shells and Utilities (continued)

- dbx enhancements (continued)
  - Problem/need addressed:
    - dbx customers need a way to debug their complex program constructs.
  - Solution:
    - Include a plug-in architecture that allows dbx users to include application specific knowledge that can be used at debug time via plug-in DLLs.
  - Benefit:
    - Provides a programmable framework to extend dbx's capabilities with unique application knowledge

Shells and Utilities (continued)

- dbx enhancements (continued)
  - Need addressed:
    - Support is required for opcodes for new instruction set.
  - Solution:
    - New support was added for the extended-immediate facility instructions in dbx.

File systems

- Addressed a number of customer requirements for ISHELL and other utilities
- New character special device files
- Enhancements to Display OMVS and SET OMVS
- Aid for HFS to zFS migration
- zFS enhancements

File systems (continued)

- ISHELL enhancements:
  - Option to specify logical or real path on the file list
  - Improve ISHELL entry messages when the user cannot access ISHELL
  - Allow specification of file attributes when creating a new file
  - Keep a path history similar to ISPF NRETRIEV (used on main panel)
  - Preserve file format and CCSID on copy

File systems (continued)

- ISHELL enhancements (continued):
  - Support a refresh command on the file list
  - Add a group list panel similar to the user list panel
  - Capture and show zFS errors when trying to create a zFS file system
  - Do not exit execute dialog until execute main panel is dismissed
  - Do not save last pathname in profile until ishell exit
  - Add filesystype selectable column to the mount table

File systems (continued)

- OEDIT:
  - Increase maximum width for file edit
  - Give warning if extended attributes are set on a file being edited before oedit causes them to get reset.
- MOUNT Utility:
  - Add a wait option (with time) so the mount will wait for async mounts to complete
- TSO Utility:
  - Allow user to allocate SYSTSPRT

File systems (continued)

- BPXWDYN:
  - Add SVC99 info retrieval capability to be able to determine the DD names and data set names, and path names for current allocations. Allocation attributes will not be supported at this time.
  - Add keys for tape processing: position, label, retpd, trtch
  - Allow resetting of S99NOMNT

- REXX
  - Change readfile to return the last line it processes when it ends in error due to line length too long

File systems (continued)

- /dev/random and /dev/urandom
  - Problem:
    - Most Unix systems support these random number generator devices.
  - Solution:
    - Add the /dev/random and /dev/urandom devices to Unix System Services.

    The /dev/random and /dev/urandom are character special device files that generate random numbers. They are open()ed and read() from like any other file. This output is used by various applications for creating security keys and other cryptographic purposes.

    On some Unix systems /dev/random may block waiting for naturally occurring randomness to occur and /dev/urandom is an alternative, less secure but non blocking, random number generator. On z/OS both of these devices are the same, they rely on the hardware to provide the random numbers, and they will not block.

File systems (continued)

- /dev/zero
  - Problem:
    - Most other UNIX platforms provide the /dev/zero device.
  - Solution:
    - Add the /dev/zero device to Unix System Services.

    The /dev/zero is a character special device file that accepts and discards anything written to it and provides binary zeros for any amount read from it.

File systems (continued)

- Display AF_UNIX Sockets and Sessions
  - Problem:
    - There is no way to tell how AF_UNIX socket programs are doing or what sessions have been established. A display function like netstat for AF_INET sockets is needed for AF_UNIX sockets.
  - Solution:
    - Enhance DISPLAY OMVS with a Sockets option that will display information about AF_UNIX Sockets and their sessions

File systems (continued)

- Display AF_UNIX Sockets and Sessions

Example output:

```
DISPLAY OMVS,Sockets|So
BPXO060I 17.12.57 DISPLAY OMVS
OMVS     000D ACTIVE          OMVS=(6F,JB)
         AF_UNIX Domain Sockets
JOBNAME    ID      PEER ID  STATE    READ      WRITTEN
--------  -------- -------- ------   -------- --------
TCPCS    00000003 00000000 LISTEN   00000345
 Socket name: /var/sock/SYSTCPCN.TCPCS
TCPCS    0000002A 00000022 ACP      000012AB 00054C2A
 Socket name: /var/sock/SYSTCPCN.TCPCS
   Peer name: /tmp/sock1
NETVIEW  00000022 0000002A CONN     00054C2A 000012AB
 Socket name: /tmp/sock1
   Peer name: /var/sock/SYSTCPCN.TCPCS
```

File systems (continued)

The fields displayed are:

Jobname - The jobname of the process that owns the socket.

Id - The Inode number of the socket, in hexadecimal.

Peer Id - The Inode number of a connected socket's peer socket.

State - The socket state, which is one of:

- LISTEN - a server TCP stream socket that accepts connections.
- DGRAM - a UDP datagram socket.
- ACP - an accepted stream socket
- CONN - a connected stream socket
- STRM - an unconnected stream socket.

Read/Written - The number of bytes read or written on this socket in hexadecimal. For a server socket the READ count is the number of connections that have been accepted. This value wraps after 4G.

Socket Name - The name this socket was bound to, if any.

Peer Name - The name of the socket this socket is connected to, if it is connected and if the peer socket has a name.

File systems (continued)

- Latch Contention Analysis
  - **Problem:**
    - When threads get hung it is difficult for an operator to determine why they are waiting. When the system gets hung it is difficult to determine who or what needs to be canceled to free things up again.
  - **Solution:**
    - Expand DISPLAY OMVS with new Waiters option. Diagnostic information displayed:
      - The **Mount Latch Tracking** display shows who is holding the Mount Latch and who is waiting for it.
      - The **Outstanding Sysplex Messages Display** shows who is waiting for a sysplex reply, what type of message was sent, to which system or systems the message was sent, and how long the reply has been outstanding.

File systems (continued)

- Latch Contention Analysis

Example output:

```
SY1  DISPLAY OMVS,Waiters
SY1  BPXO063I 12.39.07 DISPLAY OMVS 426
OMVS     000E ACTIVE          OMVS=(QY)
MOUNT LATCH ACTIVITY:
    USER   ASID     TCB            REASON                          AGE
 HOLDER:
  OMVS     000E  008E9828     Inact Cycle                       00.01.18
     IS DOING: XPFS VfsInactCall / XSYS Message To: SY2
     FILE SYSTEM: ZOS17.SY2.ETC.HFS
 WAITER(S):
  OMVS     000E  008D97C8     FileSys Quiesce                   00.00.05
  OMVS     000E  008E9B58     FileSys Sync                      00.01.10
OUTSTANDING CROSS SYSTEM MESSAGES:
 SENT SYSPLEX MESSAGES:
    USER   ASID   TCB     FCODE  MEMBER   REQID     MSG TYPE     AGE
  MEGA     0025 008DD218  0008 SY2      01000038 LookupCall   00.03.08
  TC0      0026 008E6E88  1011 SY1      0100003A Quiesce      00.00.05
  OMVS     000E 008E9828  0804 SY2      01000039 VfsInactCall 00.01.18
 RECEIVED SYSPLEX MESSAGES:
           FROM   FROM            FROM
   ON TCB  ASID   TCB     FCODE  MEMBER   REQID     MSG TYPE     AGE
  008D97C8 0026 008E6E88  1011 SY1      0100003A Quiesce      00.00.05
     IS DOING:          Mount Latch Wait-Latch 2
```

File systems (continued)

- Additional information on D OMVS,F
  - **Problem:**
    - It is often difficult to correlate the resources shown by the Contention display of GRS with the file systems that have been mounted.
  - **Solution:**
    - Expand output of D OMVS,F to include:
      - date and time of the mount
      - The File System's LFS Latch number, and its Quiesce Latch number if it has ever been quiesced by Unix System Services.

File systems (continued)

- Additional information on Display OMVS,F

Example output:

```
D OMVS,F
BPXO045I 08.11.29 DISPLAY OMVS 307
OMVS     000E ACTIVE           OMVS=(ZG)
TYPENAME   DEVICE ----------STATUS----------- MODE  MOUNTED     LATCHES
HFS             18 QUIESCED                   RDWR  08/17/2005  L=30
  NAME=POSIX.FS01                                   10.44.09    Q=31
  PATH=/tmp/new_fs
  QSYSTEM=SY1 QJOBNAME=IBMUSER6 QPID=          5
D GRS,L,C
ISG343I 10.44.37 GRS STATUS 345
LATCH SET NAME:  SYS.BPX.A000.FSLIT.FILESYS.LSN
CREATOR JOBNAME: OMVS      CREATOR ASID: 000E
  LATCH NUMBER:  31
    REQUESTOR  ASID  EXC/SHR    OWN/WAIT
    BPXAS      0023  EXCLUSIVE  OWN
    IBMUSER    0021  SHARED     WAIT
```

File systems (continued)

- Honor MOUNT commands contained in the parmlib member specified with SET OMVS=
  - **Problem:**
    - Neither SET OMVS=xx nor SETOMVS RESET=(xx) support the MOUNT command so there is no direct way to execute a list of mounts from the console.
  - **Solution:**
    - SET OMVS=xx will be enhanced to execute the ROOT and MOUNT commands contained in the specified parmlib member. FILESYSTYPE, SUBFILESYSTYPE, and NETWORK commands will also be executed.

File systems (continued)

- SET OMVS= with mount commands

  Example output:

```
SET OMVS=AV
IEE252I MEMBER BPXPRMAV FOUND IN SYS1.PARMLIB
BPXF013I FILE SYSTEM POSIX.TOTTEN.REGRESS.HFS 259
WAS SUCCESSFULLY MOUNTED.
IEF196I IGD103I SMS ALLOCATED TO DDNAME SYS00006
BPXF013I FILE SYSTEM POSIX.TOTTEN.FS 261
WAS SUCCESSFULLY MOUNTED.
IEF196I IGD103I SMS ALLOCATED TO DDNAME SYS00007
BPXF013I FILE SYSTEM POSIX.TOTTEN.HFS1 263
WAS SUCCESSFULLY MOUNTED.
BPXF002I FILE SYSTEM POSIX.TOTTEN.ZZZ WAS NOT MOUNTED.
RETURN CODE = 00000099, REASON CODE = 5BC7082A
```

File systems (continued)

- New option on Display OMVS
  - **Problem:**
    - There is no way to view recent mount or move failures from the console.
  - **Solution:**
    - Enhance D OMVS to show information about prior mount or file system move failures that have occurred.

Pertinent information from failures from prior MOUNT or MOVE File System commands, of any form, will be saved. This information will be available for display or application retrieval. This includes, for instance, mounts issued from TSO, Ishell, those from BPXPRMxx during system startup, and sysplex mounts. Also, filesystem new owner failures (move commands) that occur during setomvs, chmount, shutdown and member gone event processing will be saved.

File systems (continued)

- D OMVS,MF
  - Prints the most recent mount or move failures, up to 10.
- D OMVS,MF=all or D OMVS,MF=a
  - Prints the 50 most recent mount or move failures.
- D OMVS,MF=purge or D OMVS,MF=p
  - Purges the saved failure information.

File systems (continued)

- D OMVS,MF

  Example output:

```
D OMVS,MF
BPXO058I 09.30.57 DISPLAY OMVS 346
OMVS     000E ACTIVE          OMVS=(P1)
SHORT LIST OF FAILURES:
TIME=08.11.11  DATE=2005/08/17        MOUNT RC=0099  RSN=EF096150
  NAME=ZOS17.TMP.ZFS
  TYPE=ZFS
  PATH=/SYSTEM/tmp
  PLIB=BPXPRMZG
TIME=07.58.17  DATE=2005/08/17        MOVE  RC=0079  RSN=119E04B7
   NAME=*
   SYSNAME=SY9
```

File systems (continued)

- zFS is the preferred file system
  - **Problem 1:**
    - The need to change the file system type specification in policies and scripts.
  - **Solution 1:**
    - The file system type HFS will now be considered a generic file system type that can mean either HFS or ZFS. When HFS is specified, mount processing will look for a data set matching the file system name. If found, and the data set is not an HFS data set, the type will automatically be changed to ZFS and the mount will proceed as though ZFS were specified.

File systems (continued)

- zFS is the preferred file system
  - **Problem 2:**
    - Customers will likely want to have a different naming standard for their zFS data sets than their HFS data sets. Migration scenarios exist where either the file system names are not changed or at least file system names that contain “HFS” are changed to similar names with “ZFS” substituted.
  - **Solution 2:**
    - Capability was added to specify a file system name with substitution place holders. The place holder is /// and represents the string HFS or ZFS as appropriate, if the file system type is specified as HFS. Where /// is used, mount processing will first substitute ZFS and check to see if the data set exists and is not an HFS data set. If this is the case it will proceed with that name and direct the mount to ZFS. Otherwise mount processing will substitute HFS.

File systems (continued)

- zFS is the preferred file system
  - **Problem 3:**
    - The actual migration.
  - **Solution 3:**
    - The basic steps are
      - Define a new zFS aggregate as an HFS compatible
      - Create temporary directories
      - Mount both the HFS and zFS file systems.
      - Use appropriate utilities to copy the HFS to the zFS
      - Unmount the file systems
      - Remove the temporary directories
      - Rename the data sets as appropriate
    - To assist with the migration steps a new tool has been provided in the form of an ISPF dialog, bpxwh2z.
    - For more details:
      - *See session 2923:* ***User Experiences with HFS to zFS Conversion***
        On Wednesday 8:00 AM
      - *See session 2910:* ***Introduction to the zSeries File System (zFS)***
        On Monday 9:30 AM

File systems (continued)

- zFS enhancements:
  - Provide zFS performance statistics through a programmable interface.
    - zFS provided additional pfsctl (BPX1PCT) calls to return information on kernel calls, vnode cache, metadata cache, directory cache, log cache and transaction cache.
  - Provide an Unquiesce Aggregate Modify operator command
    - This is useful in cases where a zFS aggregate is left in the quiesced state if the job that quiesced the aggregate failed.
  - zFS file system naming convention
    - New support was added to allow all characters in the file system name that HFS allows, specifically: '@', '#', '$', and X'C0'
  - zFS command forwarding
    - Sysplex command forwarding via XCF services. zfsadm commands can now be used to display or modify zFS aggregates on any system in the sysplex. And zFS configuration parameters can be displayed or modified from any system, directed to any system in the sysplex.

File systems (continued)

- z/OS Network File Server (NFS) Version 4 of the NFS Protocols
  - **Problem:**
    - The current z/OS implementation of NFS is at the Version 3 Protocol level and the industry is migrating to Version 4 servers and clients.
  - **Solution:**
    - Provide enhancements in the Unix System Services Interfaces for Servers that are needed for NFS to support the V4 Protocols.
      - V_open and v_close - New Callable Services that provide features that are needed for the new Open and Close RPCs in the Version 4 Protocols.
        V_lockctl - Returning more information on who is blocking a lock request.
      - V_lockctl - Providing an interface to the BRLM UnLoad Locks function.
      - V_lockctl - Expanding the interface for purging locks:
        -> To purge locks held by a client user.
        -> To purge all locks held by NFSS for an object.
      - V_lockctl - Providing an asynchronous locking capability similar to that for socket I/O.
    - For more details:
      - *See session 3027:* ***The Next Generation of z/OS NFS***
        On Thursday 8:00 AM

Kernel

- Kernel support of mixed case passwords
- New facility class
- Dynamic service activation

Kernel (continued)

- Kernel support of mixed case passwords
  - Problem:
    - Other operating systems support mixed case passwords, but z/OS does not. Customers would like more consistency in this area.
  - Solution:
    - New support was added to detect that mixed case passwords are supported. This can be utilized by the security product during initialization. The kernel will continue to fold passwords to upper case if this bit is not on. When the bit is on, the folding to upper case will be bypassed. In addition to folding passwords to upper case, the Unix System Services Kernel also folded non-graphics characters to blanks. This will continue to be done for mixed case passwords.

Kernel (continued)

- New facility class profile (BPX.CONSOLE)
  - Problem:
    - In order to use the authorized functions of the __console()/__console2() services the invoker needed to be a superuser. This forced installations to give some __console()/__console2() users uid 0. This in turn gave much more authority to those users then was really needed.
  - Solution:
    - Create a new facility class profile (BPX.CONSOLE) that would allow a permitted user the ability to use the __console()/__console2() authorized functions

If the BPX.CONSOLE facility class profile is defined and the invoker is permitted to that profile or the invoker is running with an effective uid of 0, the invoker will be considered as having appropriate privileges with regards to this Service.

Kernel (continued)

- UNIX System Services Dynamic Service Activation
  - Problem:
    - Currently, customers need to completely disrupt and take down their systems to install almost all critical UNIX System Services service to their systems.
  - Solution:
    - Provide the capability to dynamically activate and deactivate service items that impact UNIX System Services component modules.

The primary objective of this support is to significantly reduce the number of planned outages by providing a dynamic service activation capability for the UNIX System Services Component. This will allow a majority of the service fixes involving the z/OS UNIX System Services Kernel and file system components to be installed and activated without requiring a re-IPL of a given system.

Kernel (continued)

- UNIX System Services Dynamic Service Activation commands
  - F OMVS,ACTIVATE=service
    - Will cause the activation of **only** those service items found in the target libraries that are identified internally by UNIX System Services as capable of being dynamically activated.
  - F OMVS,DEACTIVATE=service
    - Allows a customer to back off a set of dynamically activated service items.
  - New BPXPRMxx statements (identifying the target service activation libraries):
    - SERV_LPALIB('dsname','volser')
    - SERV_LINKLIB('dsname','volser')

Kernel (continued)

- UNIX System Services Dynamic Service Activation commands (continued)
  - SETOMVS SERV_LINKLIB=('posix.totten.linklib','bpxxxx')
    - Use to set the target library
  - D OMVS,O
    - Display all of the Unix System Services configuration settings now includes SERV_LINKLIB and SERV_LPALIB.
  - D OMVS,ACTIVATE=service
    - Displays the current set of dynamically activated service.

Unix System Services Health checks

- Verify that MAXSOCKETS (AF_INET) and MAXFILEPROC are set high enough.
  - This check will look at the values for MAXSOCKETS and MAXFILEPROC and give an exception message if either is too low. MAXSOCKETS and MAXFILEPROC values will each be compared to 64000.
- Verify that the Automount delay configuration values in a sysplex are set appropriately.
  - This check will go through all the automount configurations for a system in a sysplex and check the value of the automount delay.
  - An exception message will be displayed for every configuration with a delay time of less than 10. The message will show the automount configured directory, the configuration name and the delay value.

Publications

- UNIX System Services Planning
  - GA22-7800
- UNIX System Services Command Reference
  - SA22-7802
- UNIX System Services Assembler Callable Services
  - SA22-7803
- UNIX System Services User's Guide
  - SA22-7801-05
- UNIX System Services Messages and Codes
  - SA22-7807-05
- IBM Health Checker for z/OS: User's Guide
  - SA22-7994-00
- z/OS V1R5.0 Distributed File Service zSeries File System
  - SC24-5989