OMVS and PARMLIB

A Walk Through BPXPRMxx

Kershaw Mehta
kershaw@us.ibm.com

IBM
OS/390 UNIX System Services and Language Environment Design

Agenda

- General information
- System Commands to dynamically make changes to OMVS address space
- Go through specific descriptions
- What's new in R7, R8, R9

General Information

- OS/390 CBPDO customers - must copy BPXPRMXX from SAMPLIB to PARMLIB
- OS/390 ServerPac customers - shipped in CPAC.PARMLIB
  - BPXPRMFS - for filesystem info
  - BPXPRM00 - for other system limits
- All Customers should customize the keywords settings!!!

General Information ...

- As of OS/390 R3, need to specify OMVS=xx, where xx corresponds to BPXPRMxx, in your IEASYS member.
  - OMVS=(xx,yy,....) allows for concatenating
- If OMVS=DEFAULT or not present, then OMVS comes up in minimum mode.
  - Have taken default for system limits and no filesystems have been defined. Also the temporary filesystem (TFS) is used as root file system.

General Information ...

- Recommendation:
  - Have two BPXPRMxx members, one for system limits and other for filesystems setup. This allows for easier migration from one release to next.

- If you want one BPXPRMxx member to be shared across systems, then you can use system symbolics.
  - For Example: &SYSNAME

Operator Commands

- The SETOMVS command enables you to modify BPXPRMxx parmlib settings without re-IPLing.
  - SETOMVS MAXFILEPROC=100,MAXPROCUSER=8
- The SET OMVS command enables you to dynamically change the BPXPRMxx parmlib members that are in effect. This enables you to reconfigure a large set of system characteristics.
  - SET OMVS=(AA,BB)

Operator Commands ...

- If a parameter is specified more than once with different values, in the parmlib members, the first value specified is the first value that is used.
- For example, if you specify
  - SET OMVS=(AA,BB)
- where AA has MAXPROCUSER=10, and BB has MAXPROCUSER=5, the MAXPROCUSER=10 value is used.

MAXPROCSYS

- The maximum number of processes that can be active at the same time.
- You can manage your system resources by limiting the number of processes.
- Range: 5 to 32767
- Default: 200
- Can use SETOMVS or SET OMVS to change value as long as:
  - New value = MIN(32767,MAX(4096,3*initial value))
  - Anything higher will require changing BPXPRMxx and re-IPL.

MAXPROCSYS ...

- Avoid specifying a higher value for MAXPROCSYS than your system can support. A larger value means that more pageable storage will be allocated.

- Plan on one process for each daemon you start from a shell script like /etc/rc.
    - For example - inted and cron

- Each shell user needs a minimum of three processes and possibly a few more for piping between shell commands.

MAXPROCUSER

- The maximum number of processes that a single user (that is, with the same UID) can have concurrently active.
- A user with UID of 0 (zero) is not limited by the MAXPROCUSER value, because a superuser may need to be able to log on and use kernel services to solve a problem.
- Range: 3 to 32767
- Default: 25
- Recommended OS/390 Setting: 100
- Can use SETOMVS or SET OMVS to change value as long as:
  - New value = MIN(32767,MAX(4096,3*initial value))
  - Anything higher will require changing BPXPRMxx and re-IPL.

MAXPROCUSER ...

- To improve performance, use MAXPROCUSER to limit user activity.
- New in R8
  - For MAXPROCUSER, you can set a system-wide limit in BPXPRMxx and then set limits for individual users.
  - Use RACF ADDUSER or ALTUSER command to specify PROCUSERMAX limit:
    - ALTUSER *userid* OMVS(PROCUSERMAX(100))
  - Useful for daemons that run without UID(0) and create many address spaces. In these cases, assign the daemon user ID a high enough PROCUSERMAX value in OMVS segment.

MAXUIDS

- The maximum number of unique UIDs that can use kernel services at the same time. The UIDs can be for interactive users or for programs that requested kernel services.
- You can limit the number of active UIDs.
- Range: 1 to 32767
- Default: 200
- Can use SETOMVS or SET OMVS to change value as long as:
  - New value = MIN(32767,MAX(4096,3*initial value))
  - Anything higher will require changing BPXPRMxx and re-IPL.

MAXUIDS ...

- If the MAXUIDS value is too high relative to the MAXPROCSYS value, too many users can invoke the shell. All users may be affected, because fork calls may begin to fail.
  - For example, if your installation can support 400 concurrent processes - MAXPROCSYS(400) - and each UID needs an average of 4 processes, then the system can support 100 users. For this specify MAXUIDS(100).

MAXFILEPROC

- The maximum number of files that a single process can have open concurrently.
- Use MAXFILEPROC to determine the number of character special files, /dev/fdxx that are created.

- Range: 3 to 65535
- Default: 64
- Recommended OS/390 Setting: 256
- Can use SETOMVS or SET OMVS to change value as long as:
  - New value = MIN(65535,MAX(4096,3*initial value))
  - Anything higher will require changing BPXPRMxx and re-IPL.

MAXFILEPROC ...

- New in R8
  - For MAXFILEPROC, you can set a system-wide limit in BPXPRMxx and then set limits for individual users.
  - Use RACF ADDUSER or ALTUSER command to specify FILEPROCMAX limit:
    - ALTUSER *userid* OMVS(FILEPROCMAX(100))
  - The minimum value of 3 supports stdin, stdout, and stderr

MAXPTYS

- The maximum number of pseudo-TTY sessions that can be active at the same time.
- Use MAXPTYS to manage the number of interactive shell sessions.
- Range: 1 to 10000
- Default: 256
- Can use SETOMVS or SET OMVS to change value as long as:
  - New value = MIN(10000,MAX(256,2*initial value))
  - Anything higher will require changing BPXPRMxx and re-IPL.

MAXPTYS …

- New in R7
  - The specified value will dynamically create pseudo-TTY pair (/dev/ptyxxxx and /dev/ttyxxxx) as they are needed.
  - Lowering the value, just lower the *high water mark*.
- Recommendation: Because each user may have more than one session, we recommend that you allow four pseudo-TTY pairs for the end-user.
- MAXPTYS = MAXUIDS * 4

CTRACE

- Specifies the PARMLIB member (already defined) that contains the initial tracing options to be used.

- Can use predefined members:
  - CTIBPX00 - Allows for minimal tracing - Default
  - CTIBPX01 - Turns on all tracing

- Cannot use SETOMVS or SET OMVS to change this value. Must use TRACE command.

STEPLIBLIST

- The pathname of the HFS file that contains the list of MVS data sets to be used as step libraries for programs that have the set-user-ID and set-group-ID set.
- Default: None
- Recommended OS/390 Setting: If you are STEPLIB'ing to the Language Environment run-time library SCEERUN, you should put this data set name in this file.

- Can use SETOMVS or SET OMVS to change value.

Defining Filesystems

- Specify the TYPE of filesystem.
  - FILESYSTYPE - defines the type of physical file systems
  - ROOT - defines and mounts the root file system
  - MOUNT - defines the file systems to be mounted at initialization at given mountpoints in file hierarchy
  - NETWORK - defines information about socket physical file system
  - SUBFILESYSTYPE - identifies each physical file systems to run underneath the Common INET socket file system.

FILESYSTYPE

- Specifies the TYPE of filesystem.
  - AUTOMNT - Handles automatic mounting and unmounting of filesystems.
    - Module name - BPXTAMD
  - HFS - Needed for regular local files requests.
    - Module Name - GFUAINIT
  - NFS - Handles requests for access to remote files.
    - Module Name - GFSCINIT
  - CINET - Handles requests for the AF_INET family of sockets. Needed for running multiple TCP/IP stacks.
    - Module Name - BPXTCINT

FILESYSTYPE ...

- INET - Handles requests for the AF_INET family of sockets.
  - Module Name - EZBPFINI
- TFS - Handles requests to the temporary file system.
  - Module Name - BPXTFS
- DFSC - Enables the use of Distributed File systems running is a Distributed Computing Environment (DCE).
  - Module Name - IOECMINI

► Example:

- FILESYSTYPE TYPE(HFS) ENTRYPOINT(GFUAINIT)
- FILESYSTYPE TYPE(TFS) ENTRYPOINT(BPXTFS)
- FILESYSTYPE TYPE(INET) ENTRYPOINT(EZBPFINI)

ROOT

```
ROOT FILESYSTEM('OMVS.ROOT')
           TYPE(HFS)
           MODE(RDWR)
```

- FILESYSTEM - Name of MVS data set to be mounted as root.
- TYPE - 99.9% of the time this will be HFS
- MODE - RDWR allows read and write access
  - READ allows read only access

- Cannot use SETOMVS or SET OMVS. Can use TSO/E UNMOUNT and MOUNT commands to change root filesystems.
  - Ensure you use IMMEDIATE option on unmount,

MOUNT

```
MOUNT FILESYSTEM('OMVS.ETC')
           TYPE(HFS)
           MODE(RDWR)
           MOUNTPOINT('/etc')
```

- MOUNTPOINT - Directory where to mount filesystem

- Cannot use SETOMVS or SET OMVS. Can use TSO/E UNMOUNT and MOUNT commands to change or add filesystems.

NETWORK

- (Probably the most difficult to understand)
- To run a Single socket file system using the SecureWay TCP/IP Socket File System for network sockets:
    - The MAXSOCKET value can be changed to meet your installation requirements.

```
FILESYSTYPE TYPE(INET) ENTRYPOINT(EZBPFINI)
NETWORK DOMAINNAME(AF_INET)
           DOMAINNUMBER(2)
           MAXSOCKETS(64000)
           TYPE(INET)
```

NETWORK ...

- To start multiple socket file system with Common INET.
- Should be used if you are running Anynet and TCP/IP concurrently on an OS/390 system

  or

- If you want to isolate access from different networks to the same OS/390 system (internal company networks vs. internet access)
  - The MAXSOCKET value can be changed to meet your installation requirements.

NETWORK / SUBFILESYSTYPE

```
FILESYSTYPE TYPE(CINET) ENTRYPOINT(BPXTCINT)
NETWORK DOMAINNAME(AF_INET)
           DOMAINNUMBER(2) TYPE(CINET)
           MAXSOCKETS(64000)
           INADDRANYPORT(4000)
           INADDRANYCOUNT(2000)

SUBFILESYSTYPE NAME(tcpjob1)
TYPE(CINET) ENTRYPOINT(EZBPFINI) DEFAULT

SUBFILESYSTYPE NAME(tcpjob2)
TYPE(CINET) ENTRYPOINT(EZBPFINI)

SUBFILESYSTYPE NAME(ANYNET)
TYPE(CINET) ENTRYPOINT(ISTOEPIT)
```

NETWORK / SUBFILESYSTYPE ...

- Recommendation:

  1. The names `tcpjob1,tcbjob2,ANYNET` are the RACF userids and the name of the TCP/IP started tasks. Having the same name makes debugging connection problems easier.
  2. The first TCP/IP has been assigned as the default transport driver.
  3. INADDRANYPORT specifies the starting port number to be reserved for use by the application program that issue *port 0,INADDR_ANY* binds.
  4. You should specify a high port number for INADDRANYPORT (for example, 4000) to improve the probability that the port will be available on the transport provider.

- See *OS/390 eNetwork Communications Server: IP Configuration Guide*

MAXASSIZE

- Indicates the region size of the address space.
- This is where users who telnet into the shell get their region size. (TSO OMVS users inherit their TSO region size from logon panel).
- Range: 10485760 to 2147483647
- Default: 41943040
- Recommended OS/390 Setting:
- Can use SETOMVS or SET OMVS to change value as long as:
  - New value = MIN(2147483647,MAX(41943040,2*initial value))
  - Anything higher will require changing BPXPRMxx and re-IPL.

IPC Statements

- Keywords enable you to specify the interprocess communication (IPC) values.
  - IPCMSGNIDS - Maximum number or unique message queues in the system.
  - IPCMSGQBYTES - Maximum number of bytes per message queue.
  - IPCMSGQMNUM - Maximum number of messages for each message queue.
  - And others.....

- Recommendation: Take the default, and if you are interested in more detail, read the *Planning* book.

PRIORITYPG/PRIORITYGOAL

- List of 1 to 40 performance group numbers and corresponding names that are used when application code uses
  - nice ( )
  - setpriority ( )
  - chpriority ( )
- These functions allow a program to alter the priority of one or more processes.
- Only superusers can improve their priority values
- Regular users can only reduce their priority values.
- Default: None
- Can use SETOMVS or SET OMVS to change value.

PRIORITYPG/PRIORITYGOAL ...

- Recommendation: Generally you should not specify these in your BPXPRMxx unless you need nice( ) and setpriority( ) support. It is simplest and best to give MVS full control over priorities of work using SRM or WLM. If you still have a need:
  - Installations running in compatibility mode should use the PRIORITYPG statement to specify performonce groups. These performance groups should also be defined in your IEAIPSxx parmlib member.
  - Installations running in goal mode should use the PRIORITYGOAL statement to specify service classes. These service classes must appear in your current service policy.

New in OS/390 R8

- You can use the SETOMVS RESET command to dynamically add the FILESYSTYPE, NETWORK and SUBFILESYSTYPE statements without having to re-IPL.
  - SETOMVS RESET(tt)
    - Where BPXPRMtt is a temporary member that contains only these statements.
- However, if you change these values a re-IPL will be necessary.

New in OS/390 R8 ...

- Allow granularity on a per user basis for things like:
  - CPUTIME
  - PROCUSER
  - ASSIZE
  - THREADS
  - FILEPROC
  - MMAPAREA
- Example:
  - ALTUSER userid OMVS(FILEPROCMAX(100))

New in OS/390 R8 ...

- To shorten the names of the commands to be typed, RACF changed the name of those limits by putting MAX at the end.

- Example:
  - The ADDUSER and ALTUSER commands support CPUTIMEMAX. This allows the abbreviation of CPU instead of MAXCPU .

New in OS/390 R9

- BPXPRMxx Syntax Checker
  - You can now use the SETOMVS SYNTAXCHECK operator command to check the syntax of a BPXPRMxx parmlib member before doing an IPL.
  - You cannot use this command to verify whether FILESYS or MOUNT statements which point to modules or file system names for their validity.

- Example:
  - SETOMVS SYNTAXCHECK=(xx) - where xx is BPXPRMxx.

New in OS/390 R9 ...

- New Keywords for Shared HFS support
  - New SYSPLEX(YES or NO) default is NO
  - New VERSION('user_specified_string')
    - qualifier to represent a maintenance level of HFS
  - New parms on ROOT and MOUNT statements
    - SYSNAME declare system that is "owner" of this filesystem
    - AUTOMOVE | NOAUTOMOVE automove says: attempt to move ownership if owner fails

Reference Material

- *OS/390 UNIX System Services: Planning*
  - Detailed information on all keywords in BPXPRMxx.
- *OS/390 System Commands*
  - List of SETOMVS and SET OMVS options for these system commands.
- *OS/390 Security Server (RACF) Command Language Reference*
  - Syntax of new OMVS segment keywords.