AIX

Overview of AIX/6000 4.1

By Mark S. Brown

This article describes the new features in AIX 4.1, a major release now available.

AIX 4.1, the latest release of AIX/6000®, is a major release that includes many new features and enhancements. From the packaging of installation to the kernel-level threads support, it contains a multitude of long-awaited improvements. Even with these improvements, AIX Version 4 still maintains binary compatibility with previous versions. The only exception is a few commands, for which an easy migration process is provided.

Installation

AIX 4.1, now distributed on CD-ROM, has a smaller Base Runtime Environment install package that divides the system into modular sections. This division not only speeds up the installation process, but also enables users to customize installation of AIX 4.1 to fit their needs. As an added benefit, patches and updates are much easier to install because the number of prerequisites has been greatly reduced.

The new Network Install Manager (NIM) gives system administrators central control of installation and configuration by enabling them to install the base OS from a server onto multiple client systems via the network. With NIM, diskless, dataless, and stand-alone systems can be installed.

AIX Version 4 offers backward compatibility packages with AIX 3.2. Compatibility packages support X11R3/R4, old font types, and features such as the directory tree structure and the curses package that have been changed in Version 4.

Mark Brown

Filesystem Enhancements

When configuring Version 4, users will notice changes in the AIX Journaled File System (JFS). AIX 4.1 supports filesystems of up to 64 GB (although files are still limited to less than 2 GB in this release), BSD-style block fragments, and compression. *Fragments* are parts of a disk block; a filesystem created using fragments can store more than one file on a given disk block.

With the new JFS, a filesystem can be created from fragments of different sizes: 512, 1024, 2048, and 4096 bytes. For example, on a Version 3 JFS, a file of 500 bytes occupies an entire 4096-byte disk block. However, using the new JFS in Version 4, up to eight 500-byte files can be stored in the same disk block. The new JFS means a large savings in disk space usage, especially for users whose systems manipulate many small files.

Automatic data compression is another space-saving option in Version 4. This technique creates filesystems with a modified Lempel-Ziv technique that dynamically compresses and restores data written to disk. Users can also utilize the disk striping feature in Version 4 to set up RAID-0, which spreads the logical space of a logical volume over multiple disks. Disk striping enables simultaneous operation of multiple disks for higher performance.

Common Desktop Environment

The new look of Version 4 continues with the Common Desktop Environment (CDE). CDE is part of a co-development project among major UNIX vendors to provide portability and greater usability across platforms. X11R3, X11R4, and X11R5 are fully supported for backward compatibility.

System Management Interface Tool (SMIT)

System administrators have more help in this release through improvements to SMIT, the `mksysb` backup tool, login and password controls, and a new performance tools package. The `mksysb` tool has new extensions (such as `savevg` and `restvg`) that will save and restore user Volume Groups (VGs), save the definition of paging spaces in user as well as root VGs, and also save policy and

map files for logical volumes. SMIT also offers a non-interactive install feature.

In addition, printer support for SMIT has been greatly expanded, easing the administration of print queues. It is now possible to move print jobs from one queue to another, and to hold and release print jobs in a queue.

Open Standards Conformance

One major goal for this release was conformance to the new IEEE® POSIX 1003.2 (commands) and *X/Open Portability Guide,* Issue 4 (system interface and commands volumes). The syntax and output has been modified for some commands to meet these specifications, which will soon be met by most other UNIX vendors.

These modifications are intended to make applications more portable to and from the RISC System/6000. Version 4 conformance to XPG4 command specifications may cause some minor confusion since the syntax and output of certain commands may have changed. To minimize this confusion, IBM developers have tried to use the old syntax wherever possible.

AIX 4.1 is also a step towards compliance with the emerging SPEC 1170 Common Interface specification. As SPEC 1170 becomes final, future releases of AIX will be modified to conform to this important UNIX standard. Developers writing applications for AIX 4.1 should not be concerned since most major changes have already been made, and plans for maintaining compatibility with earlier versions of AIX are already in place.

Application Interface Changes

The curses subsystem has also been revamped, with System V Release 3 curses as its base. Developers who routinely adjust code to meet the idiosyncracies of Version 3 curses will find that the default compilation now uses System V curses as the base, with AIX Version 3 curses available to allow binary portability to Version 4. Moving application code to the new curses also ensures compatibility with SPEC 1170 curses.

A major change to the programming environment is the total rework of the threads subsystem, from the older user-level Distributed Computing Environment (DCE) threads in Version 3 to kernel-based POSIX 1003.4a Draft 7 threads. These are 1:1 (user-level:kernel-level) threads, with M:N threads intended in the near future. The OSF/1® locking model is used, and the same library mechanics (`libc_r.a`, `libpthreads.a`) are still there. The new threads provide much more function under the covers.

Other changes to the application interface include full support for the 64-bit long long `int` and 128-bit long double types, new interfaces to handle regular expressions (from POSIX), full XTI support in addition to TLI, and a full implementation of Streams. The entire tty and pty subsystems have been rewritten to take advantage of Streams.

AIX's International Language Support now encompasses 37 locales, including the bidirectional languages Hebrew and Arabic, Chinese, and many Eastern European locales. The underlying default codeset for characters (that is, the actual byte representation for the visible character) is now industry-standard ISO® 8859-1, with IBM PC-850 still provided as an alternate. Locales have been improved to include more cultural differences, such as multilevel sort capabilities. The locale conversion tool (`iconv`) has been expanded and a developer's toolkit for the Unicode™ (ISO 10646) codeset is also available. Unicode, the future direction for portable locale-independent data exchange, will be supported by AIX.

Enhancements made to the loader, linker, and binder make application development easier. Performance, efficiency in `XCOFF` size, TOC overflow handling, and `LIBPATH` handling have all been improved. Shared library data is now stored in a separate per-process shared library segment to ensure fewer conflicts for shared library data.

AIX 4.1 is a step towards compliance with the emerging SPEC 1170 Common Interface specification.

Compatibility with AIX 3.2

AIX Version 4 is intended to support, with a few exceptions, Version 3.2 binaries. Most applications written to run on AIX 3.2 will also run on 4.1 within the processor families for which they were compiled. With the advent of "common mode" compilation in AIX 3.2.5, binary compatibility extends across most of the POWER, POWER2, and PowerPC processor lines. Applications compiled for POWER processors will run on all platforms, while those compiled specifically for POWER2 or PowerPC (not in common mode) must be run only on those processors. Interface differences are also rare; a compatibility rate of over 90% is expected for existing programs.

***Mark S. Brown**, IBM Corporation, 11400 Burnet Road, Austin TX 78758. Internet: mbrown@austin.ibm.com. Mr. Brown is a senior programmer with the AIX Base Operating System Development group. He holds a BS in Computer Science from Southwest Texas State University in San Marcos.*