# AIX 3 Technology

J.C. O'Quin and C.H. Sauer

A major motivation for AIX 3 development is a significant incorporation of new operating system technology. In many areas this is based on concepts pioneered in other operating systems and prototyping efforts, e.g., in MULTICS or the CP.R project from IBM Research. However, the implementations of these concepts have still required new design efforts to provide evolutionary compatibility with existing components, interfaces and semantics. There are many areas where advances have been sought, including:

Enhanced support for, and compatibility with, relevant standards, including ANSI XJ311 (C language), IEEE 1003.1 (POSIX) and NIST FIPS 151.

BSD 4.3 compatibility.

Integration and refinement of Virtual Resource Manager (VRM) functions with the other major components of the AIX kernel, thus producing a single unified kernel.

Provision of real time capabilities, previously confined to the VRM components, to the higher level components of the system.

Provision of dynamic binding capabilities, previously confined to the VRM components, to the higher level components of the system, and support of dynamic shared libraries based on this technology.

Reimplementation of the virtual memory system and the file system to integrate the two areas. In addition, the reimplemention provides major new file system functionality:

Use of data base memory [Chang and Mergen] for file system meta-data (directories, inodes, and indirect blocks) to provide robustness and reduced need for invocation of separate consistency checking utilities, such as `fsck(8)`.

Management of disk space as "logical volumes." File systems are composed of a collection of logical volumes, possibly spanning physical volumes. File systems may be enlarged while in use by adding logical volumes. Mirroring of file systems at a logical volume granularity is implemented.

Continuation of the work to provide security characteristics as defined in the National Computer Security Center's *Trusted Computer Systems Evaluation Criteria*.

---

AIX is a trademark of International Business Machines Corporation.
UNIX is a registered trademark of AT&T.
The X Window System is a trademark of Massachusetts Institute of Technology.,
Address correspondence to Charles H. Sauer, IBM Advanced Workstations Division, 11400 Burnet Road, Austin, TX 78758. email: @cs.utexas.edu:sauer@ibmaus.uucp.

89TH0281-6/89/0000/0096$01.00 © 1989 IEEE

Advancement of distributed system capabilities, including support for internet protocols, implementation of location transparent access to memory mapped files, provision of enhanced, configurable authentication/authorization mechanisms, incorporation of Apollo Computer's object oriented remote procedure call facilities (NCS) and implementation of transparent remote execution facilities.

Provision of integrated system management facilities for both the newly developed components and existing system components.

In addition to these functional characteristics, a major objective of the new designs and the redesign of existing components is portability, so that the new system is supportable on a variety of hardware. There is inherently a tension between function, performance and portability, so "portability" has been traded off in some areas, e.g., some of the virtual memory *implementation*, with the expectation that the virtual memory *interfaces* would be portable.

Given the breadth of the above ambitions, it is impractical to try to describe all of the characteristics and technologies incorporated in AIX 3. The remainder of the paper will clarify some of the above topics. Others, e.g., the integration of the file system with virtual memory and the use of data base memory, have been discussed in other papers. Other topics will be left for future discussion.

## ENHANCED SUPPORT OF INDUSTRY STANDARDS

Just as there is a tension between advanced function and portability, there is a tension between advanced function and standardization. In the current open systems environment, conformance to standards is mandatory. Advancement of standards, e.g., resolving conflicts between *de facto* standards, may also be considered technological advancement, both because of the enabling of other technologies and because of the inherent difficulties in resolving the conflicts.

### BSD 4.3 Compatibility and Completeness

The BSD implementations of the Unix operating system have been one of the major sources of innovation and richness. Prior to AIX 3, AIX had incorporated a variety of functions originally developed at Berkeley. However, AIX was not strictly compatible with BSD, nor was it complete with respect to the body of BSD functions. One of the major efforts in AIX 3 design was the provision of strict BSD compatibility. This is a non-trivial goal given the many differences (and conflicts) between the BSD releases and the System V origins of AIX. A major effort was untertaken to define a "convergence" of BSD 4.3 and AIX [Sauer *et al* 1989]. The objective was to makethe resulting system feel like a BSD system for both end users and programmers.

## KERNEL INTEGRATION

The Virtual Resource Manager component of the AIX 1 and AIX 2 systems provides functions either not previously present in systems based on versions of the Unix operating system or not present at the level of capability of the VRM implementation:

Virtual memory management which allows for the kernel itself to be paged, and allows for the major virtual memory data structures to be paged.

Virtual terminal management which allows for multiplexing of console display and input devices.

Preemptive priority scheduling of processes, even while executing in privileged mode.

Dynamic binding of device driver code and some other operating system components, while the system is running.

(See [Loucks and Sauer 1987] for more discussion of VRM characteristics.) However, the strict boundary between the VRM and the rest of the AIX 1 and AIX 2 systems holds significant disadvantages:

Device driver code is necessarily separated into two portions, not necessarily in a natural fashion, even where it could otherwise be naturally unified.

Process scheduling/dispatching is carried out at two levels, for VRM level processes and for application level processes. Besides adding complexity and overhead, this adds latency to application level process dispatching.

Important VRM capabilities, e.g., dynamic binding and real-time support, are not available outside of the VRM.

Depending on one's point of view, the AIX 3 kernel consists of a traditional kernel enhanced to provide the functions originally in the VRM, or the VRM code extended to assume the entire kernel responsibility. Though there is some legitimacy to each point of view, the relevant technical issues are largely independent of this distinction:

Compatibility with old interfaces. In so far as practical, traditional interfaces used by device drivers and similar components are preserved.

Adaptation to preemptability in "kernel" mode. Much of the original kernel code assumes that once a process enters the privileged code of the kernel, it has effectively locked the kernel data structures. Though interrupts will be serviced, the actions performed by an interrupt handler are relatively constrained, and other processes will not get control of the processor until the process in kernel mode effectively relinquishes the lock, by either leaving kernel mode or invoking the sleep() function. This full kernel lock strategy can be a serious limitation in responding to "real time" or other high priority demands. AIX 3 allows preemption of processes in kernel mode. Locking is typically performed at a finer granularity. However, the locking services allow for the degenerate case of emulating the traditional full kernel lock semantics. Thus existing code can be ported to the preemptive environment without requiring the level of locking analysis required in the general case.

## DYNAMIC BINDING/SHARED LIBRARIES

Until recently, most binding and shared library implementations in Unix operating systems have been relatively static, e.g., as compared from the dynamic characteristics of

MULTICS. AIX 3 is designed to use a new object module format, based on extensions to the COFF format, to enable dynamic binding of unresolved symbols at load time and, if demanded, during program execution. An exported symbol strategy is used to avoid the potential overhead of resolving arbitrary entry points. System calls themselves are symbols exported by the kernel and dynamically bound at `exec()` time. The mechanisms used are very similar to those previously used and documented for dynamic binding of device drivers and other code in the VRM. However, with the new design, these mechanisms are available not only for kernel level code, but also for user level code. These mechanisms also enable a fairly dynamic shared library design, allowing for relatively simple replacement of library components.

## LOGICAL VOLUMES

Rather than the rigid partitioning previously used to manage space allocation on disk, AIX 3 is designed to use file systems consisting of an extensible number of "logical volumes." Logical volumes are intended to be on the order of several megabytes of contiguous disk space. A file system may span physical disk boundaries by consisting of logical volumes on different physical volumes. The design also supports mirroring at the logical volume level.

## DISTRIBUTED SYSTEM SUPPORT

As part of the AIX 3 development effort, we are extending our Distributed Services technology [Sauer/Johnson *et al]* in the following areas:

Portablility, including portability to operating systems not based on the AIX operating system or the Unix operating system.

Extension and generalization of the concepts in DS 1, including the following additions in particular,

Support for remote mapped files. (This is designed to be optional in the sense of allowing systems to not support mapped files at all and still be more or less full participants in the network.)

Support for configurable authentication and authorization models. These include *Kerberos* based models and others.

Support for configurable network transport protocols.

Provision of new kinds of functions not previously addressed in DS 1, including remote procedure call and remote execution facilities.

## SUMMARY

AIX 3 development has ambitious goals. It strives for strict upward compatibility with formal and *de facto* standards related to the Unix operating system. It also strives to incorporate major elements of operating system technology, both old and new, not nor-

mally present in Unix operating system implementations. Finally, it is intended to be a base for further evolution in operating system development, including offerings from the Open Software Foundation.

## REFERENCES:


A.Chang and M.F. Mergen, "801 Storage: Architecture and Programming," *ACM Transactions on Computing Systems 6*, 1 (February 1988).

S.R. Kleiman, "Vnodes: An Architecture for Multiple File System Types in Sun UNIX," *USENIX Conference Proceedings*, Atlanta, June 1986.

L.K. Loucks and C.H. Sauer, "Advanced Interactive Executive (AIX) Operating System Overview," *IBM Systems Journal 26*, 4 (1987).

L.K. Loucks and C.H. Sauer, "Saving the Day," *Unix Review 7*, 1 (January 1989).

C.H. Sauer, K.A. Bohrer, T.G. Lang, C. Minshall, G.L. Owens, K. Solem and B.J. Walker, "Convergence of AIX and 4.3BSD," *UniForum Conference Proceedings*, San Francisco, February 1989.

C.H. Sauer, D.W. Johnson, L.K. Loucks, A.A. ShaheenGouda, T.A. Smith, "Statelessness and Statefulness in Distributed Services", *UniForum Conference Proceedings*, Dallas, TX, February 1988.