Patricia Seybold's

VOL. 4, NO. 2
ISSN: 0887-3054
FEBRUARY 1989

UNIX IN THE OFFICE

PRODUCTS • TRENDS • ISSUES • ANALYSIS

A Tale of Two Operating Systems

System V.4 and OSF 1

By Judith S. Hurwitz

IT IS INEVITABLE that the road to standards will not always be smooth. The Unix industry over the past two years has passed through periods of exhilaration at the idea of a unified Unix operating system, destined to bring together all the various brands into a single, cohesive operating system that would meet everyone's needs. It seemed like a doable task, and, with much fanfare, industry experts started work on it. *(continued on page 3)*

• E D I T O R I A L •

User Interface

OSF chooses a combination of technologies, leaving the door open for innovation.

By Judith S. Hurwitz

THE WORD IS OUT. OSF (Open Software Foundation) has announced its user interface selection, which it calls Motif. We think that the choice was a good one. It meets some of the most important criteria set by the leading vendors in the Unix industry. First, it maintains consistency with the PC world through the use of the look and feel of Presentation Manager (PM). This technology, offered by Hewlett-Packard and Microsoft, should help bring Unix into the mainstream: If Unix applications look and behave just like PC applications, how bad could they be? Over the next few years, users will expect applications to have a windowed interface with the characteristics and behavior of PM. It is to Unix's benefit to follow this trend and make hay.

The second most critical component is the part that users won't see, the underpinnings that developers will use to make it all work. The tools developed by Digital for its DECwindows product are a rich set of development tools and APIs. Digital has put an incredible amount of time and effort into its interface product. The effort shows, and OSF was smart enough to take advantage of Digital's technology offer. The third component—and perhaps the most creative—is that OSF left the door open for new ideas that aren't quite ready for Motif. This catalogue of technology will include some select pieces that have potential enough to make OSF try to include it in future generations of products. For user interface, for example, OSF has selected Apollo's Open Dialog presentation and dialogue manager; Base/OPEN, a user interface management system submitted by Swedish Telecom Group; and Generic Window Manager, a user interface management system submitted by Groupe Bull. A third research program adds a set of products that, while not yet compatible with OSF's core offerings, are being considered for future integration. The first such product is Carnegie-Mellon's Andrew Toolkit.

What impresses us the most in the OSF selection process is this combination of technologies. OSF resisted two potentially dangerous scenarios: It did not take one complete submission, and it did not take too many small pieces from too many places. Had OSF chosen only DECwindows, for example, the organization would have been accused of deal-making. An amalgamation of seven or eight technologies would have been interpreted as a hodgepodge that would never work in the real world. By dangling other interesting technology and research, OSF achieves an important organizational goal: keeping the door open to the addition of advanced technology—when it is economically and technically feasible.

Perhaps this competition to be the best in a timely manner will be good for the Unix community. OSF team members worked night and day to make sure the promised deadline was met for announcing their user interface.

Now OSF will face its next challenge—getting software out to the marketplace and into the hands of commercial users. We think that OSF's decision to put user interface onto as many platforms as possible is a wise one. After all, in the long run, no one will talk about the stuff that kernels are made of—they'll talk about how easy and intuitive the applications and user interface are. ◐

Patricia Seybold's Office Computing Group

Publisher PATRICIA B. SEYBOLD

Editor-in-Chief JUDITH S. HURWITZ

Managing Editor
RONNI T. MARSHAK

Senior Editors
MICHAEL D. MILLIKIN
JUDITH R. DAVIS
Telephone: (617) 861-3926

Associate Editor
JOHN N. KELLY

News Editor
DAVID S. MARSHAK

Contributing Editor
GARY J. NUTT
Telephone: (303) 444-0341

Assistant Editor
LAURE BROWN

Sales Director
RICHARD ALLSBROOK JR.

Circulation Manager
DEBORAH A. HAY

Customer Service Manager
ANNE L. WORKMAN

148 State Street, Suite 612, Boston, Massachusetts 02109 Telephone: (617) 742-5200 FAX: (617) 742-1028

Patricia Seybold's UNIX in the Office (ISSN 0887-3054) is published monthly for $495 (US), $507 (Canada), and $519 (Foreign) per year by Patricia Seybold's Office Computing Group, 148 State Street, Suite 612, Boston, MA 02109. Application to mail at second-class postage rates is pending at Boston, MA and additional mailing offices. POSTMASTER: Send address changes to *Patricia Seybold's UNIX in the Office*, 148 State Street, Suite 612, Boston, MA 02109.

•OSF•

(continued from page 1)

But this age of innocence has passed, and we are now moving into a new age of experience as Unix begins to tackle its next challenge: making it in the commercial arena. The complexity is beginning to show through the veneer. We are confronted not with the possibility of a single variety of Unix but with two competing versions of a standards-oriented operating system. The success of Unix will not be easy, and it will not come cheap. The evolution of two new organizations, the Open Software Foundation (OSF) and Unix International, is an indication of how much is at stake.

The industry cannot go backwards. The goals of OSF are very important, and so are the goals of Unix International. Perhaps we need both organizations so that each can keep the other honest.

The prevailing mood within the computer industry today seems to be that we should return to the good old days when all we had to worry about was AT&T and how fast it could be pushed to move the operating system into the 20th century. And with the company's conversion from proprietorship to openness, the industry was ready to forget all the problems of the past years. Perhaps OSF was merely temporary adolescence. Now that AT&T and Sun had grown up, there was no need for this thorn in anyone's side. Let's get back to the business of Unix, the crowds were heard to shout (silently, of course).

Our belief is that the industry cannot go backwards. The goals of OSF are very important, and so are the goals of Unix International. But perhaps, in this critical time for Unix, we need both organizations so that each can keep the other honest.

On the surface, the situation seems like a fight over control of an operating system, which, indeed, it is. Powerful companies like IBM, Digital, and Hewlett-Packard, to name a few, are accustomed to controlling their own destinies. The idea that AT&T and Sun would have total control of their operating systems was too much for these powers to stand, especially when they began to realize that Unix would continue to increase as a percentage of their business.

Other vendors used Unix as their standard operating system for years before it became fashionable. Take NCR, for example. When this pioneer introduced the Tower X years ago, it only whispered the fact that Unix was the operating system inside. The management empowered to create one company out of Burroughs and Sperry realized that, without the help of standards and Unix, it would be almost impossible to become Unisys. That story is repeated over and over again. How did these companies react to the formation of OSF and the potential that AT&T would lose a power struggle? They were frightened. Would customers perceive that things were unsettled and it would be best to postpone purchases until the industry settled down? Would they be left out in the cold with an operating system and incompatible software? Many of these organizations were suspicious of the motives of IBM in particular. Why had OSF selected IBM's operating system so quickly? Wasn't IBM getting a financial windfall? After all, OSF may have simply been a plot to slow the pace of Unix.

It is pointless to talk about the heated words exchanged between the two groups. More has been said about this than about any other occurrence in the computer industry, including the development of the Mark 1. The real point is that there will be two operating systems surfacing over the coming months from OSF and Unix International. Each group will be working hard to convince Unix buyers that its operating system is better, pointing to one feature or another as evidence.

But, ironically, neither System V.4, the forthcoming version of Unix from AT&T, nor OSF 1, the initial OSF operating system based on IBM's AIX Version 3, provides all that the commercial Unix user needs. Both have moments of brilliance. Both have characteristics that will lead them in the right direction. But various, albeit esoteric, operating systems have had all these new, exciting features and a lot more for the past years. Therefore, although we would like to report that both operating systems are the most advanced technology the industry has to offer, we cannot. Unix is being dragged, kicking and screaming, towards becoming a commercial standard. It never was built that way. AT&T never had a reason to view this strange and wonderful operating system as a source of the future of commercial computing. Some of the more critical problems that have plagued Unix over the years have been solved, but detractors have used them as proof that Unix is unsuitable for commercial use. In some ways, the detractors were not wrong to point to chronic problems within Unix. As different vendors tinkered with the basic operating system, adding bells and whistles, they often failed to correct the problems that have been eliminated in newer versions of Unix. So these problems continued to frustrate users for years longer than necessary.

Ironically, even when it seemed that Unix might be more important than anyone dreamed possible, AT&T's research organization was not used to moving quickly with technological innovation. Therefore, changes have come slowly to its operating system. In fact, the slow pace allowed Sun Microsystems, with its brilliant and aggressive development team, to leapfrog over AT&T, and was also the reason so many variations of Unix were developed. System V was the beginning of a good idea, but it could not be used as is. Developers had to add value to it before it was fit for an end user.

Despite the politics and the divergent directions that vendors are taking their operating systems, it is clear that System V.4 and OSF 1 will be the basis for what the Unix community will be living with for many years to come. Let's now take a look at each of these operating systems and see what the two groups are doing. One caveat: Both operating systems are still incomplete, so there will be some changes by both organizations even before this article makes it into print. But we'll take a snapshot of what things look like today. We will, of course, update you as things change—and they always do.

System V.4

System V.4 is a hybrid, a superset of System V.3.2, so much of it will be familiar to those used to System V.3 operating system technology. Not surprisingly, a lot of Sun OS also shows up in System V.4. In fact, most of the advanced features of System V.4 come from Sun OS. In addition, the work to move Xenix into mainstream Unix has been completed through a 386 version of the operating system.

System V.4 is a hybrid, a superset of System V.3.2. Not surprisingly, a lot of Sun OS also shows up in System V.4. In fact, most of the advanced features of System V.4 come from Sun OS.

The new operating system is being billed as "Unified Unix." Is this a concrete technological leap or a public relations move? Probably a little of both. AT&T and Sun have made it possible to incorporate features of Sun OS, BSD, and Xenix into the new operating system, though the integration is by no means tight. Critics charge that the four operating systems have been patched together. But, from a political perspective, it was a brilliant move. Who could argue with the potential of a single version of Unix that brought together the top versions of Unix into a single operating system?

Under these circumstances, it is not surprising that System V.4 is evolutionary, not revolutionary. Much of the work required to prepare it has been taken up in providing migration tools from these various operating systems to the new version. First, we will examine key functionality.

The Kernel

The structure of the kernel for System V.4 is very similar to the kernel for System V.3.2. There is some movement to begin increasing modularity, primarily implemented in the streams mechanism. The differences include the addition of BSD, and Xenix commands and system calls. Calls that for one reason or another were not added to the kernel have been placed in a compatibility package. Therefore, users can work with a command or call that is not included as an official part of System V.4 via the compatibility package.

Above all, AT&T's goal is to make the transition from System V.3 to System V.4 as painless as possible. In fact, some disappointment with System V.4 comes from the fact that AT&T felt it had to be so cautious. We would have liked to see a few more changes to advance the commercial aspects of the operating system (like disk-mirroring). In its defense, AT&T has an obligation to concentrate on maintaining compatibility with older versions of the operating systems.

Another change is that the kernel has been made preemptive. It is one of the characteristics found in some operating systems that support multiprocessors.

Virtual Memory

System V has a new virtual memory system called VM, which has been adapted from the Sun OS Release 4 and includes support for memory-mapped files. This memory mapping allows users to access file system data in a much simpler way than that previously allowed under System V. With this feature, the user can map part of a file into the address space rather than bringing data into buffer memory, which makes for more flexibility, portability, and a more efficient use of resources. In essence, this new memory mapping provides a named memory facility and allows processes to map files or devices into their memory spaces. Once mapped, the contents of files can be accessed as memory locations.

SINGLE-LEVEL STORE. System V's memory-mapping also allows physical memory to be constructed as a single-level store. Single-level store allows for more economical use of physical memory by treating all user memory as cache, eliminating the use of buffer cache for file I/O. Essentially, it is a unified mechanism for accessing file system data. In contrast, programs executed in a traditional Unix operating system like BSD are mapped to address space and then automatically brought from system into disk page by page, requiring repeated and memory-intensive I/O calls. In traditional Unix, it is difficult to fine-tune the use of memory because of the split between memory used for I/O and memory used for programs.

SHARED MEMORY. While shared memory has existed in System V for some time, it has been merged with Sun's implementation.

DYNAMIC LINKING. For a long time, users have been asking AT&T to add the ability to dynamically link subroutines. This facility is important because it provides flexibility in program development cycle, compile debug, and recompile. Like many of its other new features, System V's dynamic linking facility also comes from Sun OS. Dynamic linking is especially important for low-end systems because it allows the user to automatically shift pages into and out of memory.

File Systems

Whenever proprietary operating systems bigots want to poke holes in Unix, they point to the weaknesses in the file system. Some of these criticisms are justified. The native System V file system was designed to treat all files equally and is not modular. Therefore, replacing one component is tricky. However, other problems were bugs detected and corrected many years ago. One key problem with the current file system is the way users are forced to deal with change. For example, to change size of a local volume, the user is forced to dump the files to tape and rebuild them.

AT&T now views file system technology differently. Its developers have realized that users want to choose among several file systems. As object-oriented file systems are developed, users clearly want to be free to replace the default file system with new technology.

System V.4 allows users to choose from various file systems, including AT&T's own Remote File System (RFS), Sun's de facto standard Network File System (NFS), and BSD's popular Fast File system. To support multiple file systems, System V.4 has implemented a Virtual File System (VFS) as a means of organizing all file system operations. The VFS is a merger of the traditional System V File System Switch (FSS) and the Sun OS VFS mechanism. To achieve this file system independence requires the use of a Vnode mechanism. The Vnode, in effect, acts as a master switcher, allowing the user to select among file systems. AT&T intends to publish the VFS kernel interfaces so that third parties can allow new file systems to be implemented under System V.4.

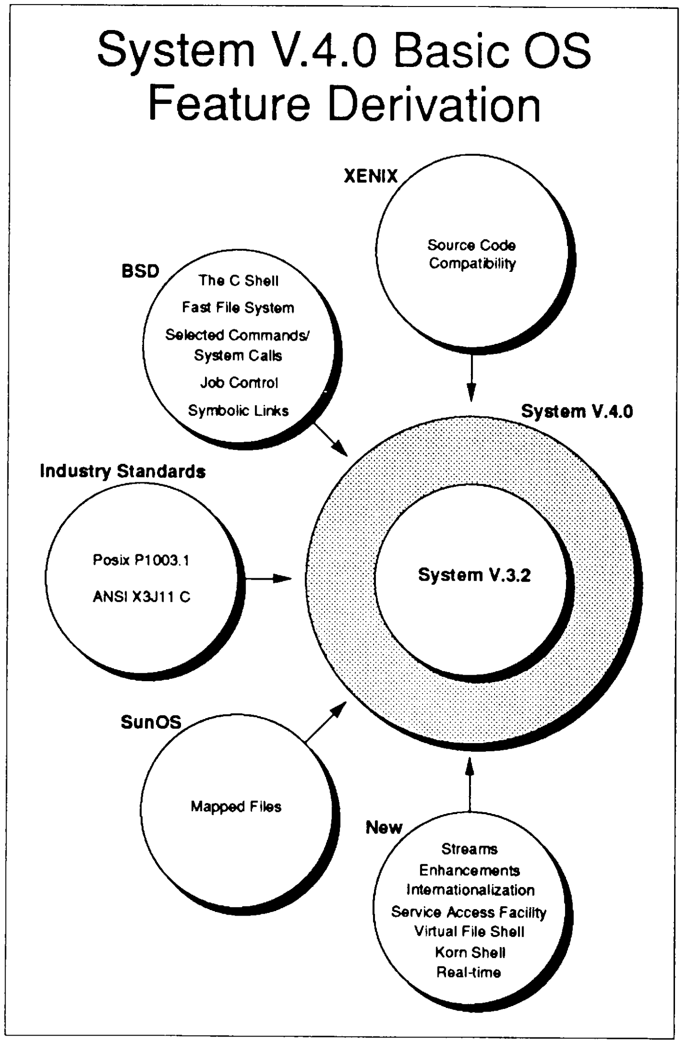
RELIABILITY. Some important improvements have been made to file system reliability. For example, bad blocks are now handled dynamically. System V.4 also includes a deferred-write scheme for modified files. This method of implementing file I/O is a critical component in commercial grade transaction processing systems and database managers since it ensures that new information is not written to disk in an incomplete form. Therefore, once the system has been recovered after a crash, the information should be consistent. However, before true commercial quality is implemented, the system would need to have roll-back and recovery features. While System V.4 has a write-through option, disk-mirroring has not yet been implemented.

FILE-LOCKING. Another significant improvement is the way System V implements file-locking. The key changes include the addition of advisory and mandatory file- and record-locking, synchronous Write mode, and Xenix file- and record-locking compatibility.

POSIX MANDATED CHANGES. The IEEE Posix specification is beginning to be implemented. For example, System V.4 has added file renaming, file truncation (previous versions of System V had no ability to truncate a file), file synchronization, ENAMETOOLONG error (an error message meaning a user has input a name that is too long), and NONBLOCK mode.

OPEN FILES. One benefit of the work being done to modularize the System V.4 kernel is that memory will be allocated dynamically. The number of open files per process has been increased from a limit of 20 file descriptors per process to 64. However, this default can be overridden through a tunable soft limit of up to 2,048 open file descriptors. The System V.4 standard I/O package allows a program to access as many as 256 open files, a limit required for binary compatibility. This type of flexibility is critical for creating an operating system that needs to scale from low-end 386 systems to mainframes.

LINKING. System V.4 has added support for symbolic linking supported in the BSD operating system. Earlier versions of System V only supported hard links. Symbolic links are most important in the network computing environment because they increase the efficiency of disk utilization by cutting down on the number of copies of files stored. Symbolic linking allows different nodes on the network to share files. Symbolic links serve as pointers, allowing a file to be stored on a server.



PROCESS FILE SYSTEM. A new facility in System V.4, the "/proc file system," generalizes the file system concept. This facility allows Unix processes to look like files and, thus, have names. It makes the debugging process easier because programmers can access processes by name as if they were simple files, rather than using special facilities as they did under earlier operating systems.

Scheduling and Real-Time Processing

The need to add real-time capabilities to Unix has been in the forefront of the debates on the commercial applicability of Unix. Given the traditional Unix round-robin approach to scheduling, it is an area of genuine concern. True real-time is really necessary only in critical process control applications, but the predictability of response is what most commercial users require.

A key issue for real-time is the latency time for process switching. Preemptive pointers have improved real-time in System V.4. A preemptable kernel is highly modularized so that machine dependencies are isolated. How does this involve the scheduler? System V.4's scheduler supports two classes of processes: one for traditional time-sharing and another for real-time or absolute processes. If the scheduler is sent a message that an application has highest priority, then the real-time process is invoked. With the real-time process, the system call "priocntl" can modify the scheduling class, priority, and minimum time-slice; priocntl is restricted to an effective user ID of zero. The real-time scheduler module follows the BSD time/timer manipulation interfaces.

System V.4 has not implemented multiprocessing. Developers we talk to insist that the first step towards multiprocessor support is making the kernel preemptive.

Multiprocessor Support

The ability to support multiprocessors is becoming an important requirement in operating systems in general, although it has long been available under certain specialized operating systems environments. In recent years, because of competitive pressures to pack more horsepower into systems, Unix vendors have used a variety of approaches to simulate multiprocessor configurations under Unix. A few, including Honeywell Bull, Arix, and Data General have done the most advanced work in Unix to date. Vendors have had to implement multiprocessing by adding layers of software on top of the operating system.

System V.4 has not implemented multiprocessing. Developers we talk to insist that the first step towards multiprocessor support is making the kernel preemptive. How soon System V will support multiprocessing is unclear. To a large degree, it will depend on how much pressure the newly formed Unix International puts on AT&T to do the work.

Internationalization

Of all the operating systems, Unix comes closest to an international operating system. A push is coming from Europe and especially from Japan to make Unix comply with established international requirements. System V.4 has made a good start. For example, it allows full eight-bit character support. As expected, it includes compliance with X/Open, ANSI, and Posix requirements such as wide-character handling, setting locate, and messaging.

Communications and Networking

STREAMS. Since its introduction as part of System V.3, Streams has been viewed as a major improvement over traditional Unix I/O mechanisms. Prior to the development of Streams, any time a new device driver was developed, new kernel software would have to be written to interface to that device. In effect, adding a new device required that the Unix kernel be rebuilt, which was no small task. Therefore, it is easy to understand why Streams was heralded so widely as an answer to a programmer's dream. Streams is a framework for character I/O. It modularized the kernel for I/O so that a set of standard interfaces could be placed within it, making an interface between each device driver and the kernel unnecessary. The added benefit (and perhaps the most important for distributed network computing) is that Streams also hides the network protocol and media, and enables a program to link to resources across the network transparently. Therefore, Streams isolates applications from networking details. Whether an application requires the use of the X.25 protocol or TCP/IP is hidden from the developer.

One important change to Streams is the addition of Streams-based pipes and named Streams facilities. Also, a TTY subsystem in the kernel has been rewritten to use the Streams mechanisms. This helps to unify the interfaces that a program uses to communicate with character devices and other processes.

One of the most important facilities that Unix system V.4.0 provides for protocol and media independence is TLI (Transport Level Interface), a critical component of Streams. Any network conforming to the Transport Provider Interface spec can be accessed by a program using TLI.

INTERPROCESS COMMUNICATION. IPC mechanisms supported in System V.3.2 are all supported in System V.4. Several additions come from Xenix, including Xenix Semaphores and shared data. BSD sockets (features of 4.2 and 4.3) have also been added. TCP/IP has become a de facto communications standard for Unix, and it is now officially part of System V.

NFS SUPPORT. Sun's de facto standard Network File System (NFS) has been added, as has Sun's External Data Representation. Sun's Yellow Pages (YP), a distributed data lookup facility added to help make NFS gain the advantages of statefulness, is supported. Without the help of auxiliary processes, NFS is stateless. This means that the client carries its own file structure and has no relationship to the server, and the server doesn't keep track of any past requests. Therefore, on the positive side, if a client or server should crash, the client simply resends the data to the server. One ramification of statelessness is that it precludes file-locking. But Sun's solution to NFS's statelessness—the addition of a lock manager and a directory manager (Yellow Pages)—slows operations. Because of these problems, AT&T developers expect that new services in the future may supersede the need for Yellow Pages.

While AT&T contends that its own network file system, RFS (Remote File System), has been integrated with NFS, in reality, the two share an administrative menu. Another facility allows files to be transferred between the two RPCs.

NEW NETWORKING FEATURES. New facilities introduced in System V.4.0 include a major revision to uniformly support the many different underlying mail architectures found on Unix systems. The enhancements will maintain compatibility with ATTMAIL and DARPA mail systems, and will handle binary messages as well as the conventional text messages. Part of the changes to mail will allow for name-to-address translation for handling the network addresses of servers in a transparent manner.

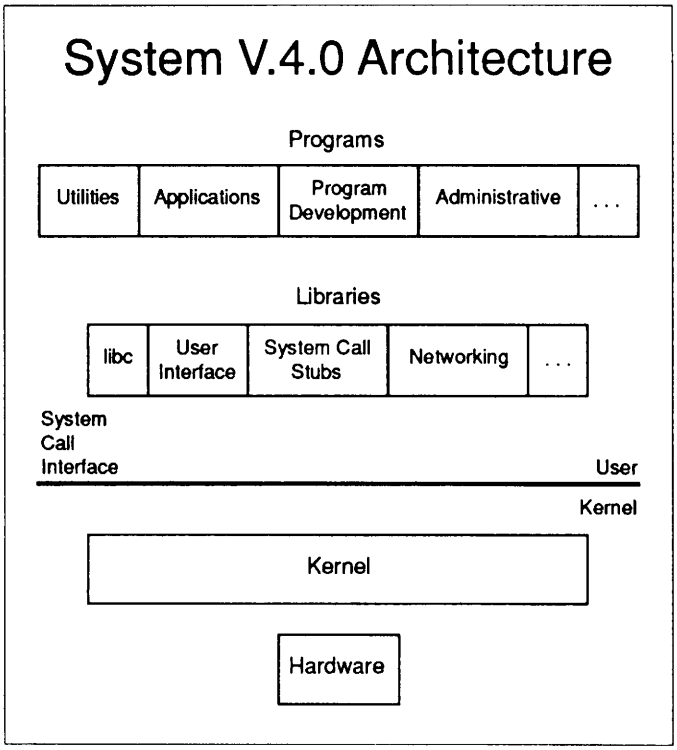
Networking issues not supported in System V.4 include OSI and mainframe or PC connectivity. It may well be that with all the work to be done, developers decided to leave connectivity to third parties—at least for now. However, if Unix is to truly go commercial, this issue should take higher priority.

UUCP has been upgraded to allow for access via higher speed networks such as LANs. Other new features include automatic recovery from partial transfers, job grades, improved administration, better logging and auditing capabilities, and more control over job grade and resource consumption.

Shells and Interfaces

User interface has become one of the hottest topics in Unix today. It is clear that the debate will not end soon. Initially, the only way to construct an interface for end users was via shells. Shells are, in fact, very powerful, not only for creating menus but also for creating cross-application macros. Releasing the power of macros, System V.4 has expanded the supported shells to include de facto and popular shells such as the C Shell and the Korn Shell.

The problem of end-user interface is not as easily solved. When AT&T first announced that it would develop a combined operating system with Sun Microsystems, it included Open Look user interface as the standard. However, AT&T is feeling somewhat burned after the industry uproar over its partnership and has backed off. Although it is still pushing hard for acceptance of the interface, the company has not yet decided if Open Look will be officially part of the operating system or will be sold as an option. We believe that AT&T is waiting to see which user interface or interfaces will emerge as de facto standards.



OPEN LOOK. Open Look is a visually pleasing user interface. It is graphically designed and includes many innovative features. For example, it uses the push-pin metaphor for invoking an option and includes many of the characteristics of other new-generation user interfaces, such as pull-down menus. However, it also allows users to access submenus at different physical points on the screen. Developers also spent time making error messages very graphical by displaying them in 3-D boxes.

There are three character-oriented libraries dealing with forms, menus, and panels that compose the Extended Terminal Interface (ETI). Two libraries involve the X-Window system Version 11.2 libXt and libNDE (NeWS Development Environment library). Both are bit-mapped interface libraries on top of which the Open Look look and feel will be implemented. The networking libraries include libnet network selection and listener services, lisnsl-network services, library TLI interface, librpc-remote procedure call, and libxdr.

While much of the technology being developed for Open Look is a good beginning, there are a few areas of concern. First, a considerable amount of work remains to be done. It is possible that all aspects of the interface will not be completed when the operating system is released to end users in the third quarter of 1989. For example, when we last saw Open Look, the graphically pleasing error messages caused other applications to freeze. Also, developers have not decided how to handle National Language Support within the Open Look environment. Therefore, it would be impossible for the user to move out of the error message to take a look at the source of the problem. Second, there is some question about how widely this interface will be accepted. If, for example, users want a user interface that at least looks like Presentation Manager, will they accept Open Look? We expect there will be some confusion between Open Look and the merged version Of Sun's NeWS/X.11.

Administration

AT&T has begun to enhance administration by adding a front-end menu. In addition, there are aids for system installation and removal, configuration management, backup, and restore. There is also a user interface that merges administrative functions for NFS and RFS.

Auditing and Security

System V.4 supports C1-level security. This will be a disappointment to users looking for more secure Unix. Most federal government procurements require a B2 security level. C would be too low for many requests for proposal. In System V.4.1, AT&T intends to implement B2-level security.

In addition, no change has been made to the super-user in System V.4. The AT&T developers we spoke to are working on a different approach to the thorny super-user issue for future versions of System V. For example, security will follow the principle of least privilege. Therefore, it takes on the look of a building with many rooms, each requiring a separate key. A master key would still be needed, but the possibility exists for several masters, each one fitting a set of doors. This next-generation security system will be designed so there is less chance of people breaking in.

There is probably some truth to the assumption that OSF chose AIX to ensure that IBM would come on board. However, we don't think that it was a direct demand from IBM for participation.

Developers' Tools

One important change in System V.4 for developers is the replacement of the Common Object Format (COF) with the Extended Linker Format (ELF). An object format defines how the binary and object files are formatted. This change is particularly relevant for programmers writing compilers and loading different binaries together and for developers who write linkers and debuggers. ELF, the object format from Sun OS, is more suitable for high-level languages such as Ada or Cobol, which COF doesn't support. However, because COF has been widely accepted and used for the C programming language, the large number of utilities that have been developed around it will have to be rewritten to support ELF. In addition, Unix C programs will have to be recompiled to support the new format. AT&T promises to continue to support COF and will provide migration tools to ELF.

OSF 1

OSF 1 will be the first release of the Open Software Foundation operating system. It is based on a forthcoming Version 3 of IBM's AIX. Although other functions, such as user interface, multiprocessor support, etc., were left open for future debate, the operating system was not. From the inception of OSF, AIX has been chosen as the base operating system. Some skeptics have proclaimed that AIX was chosen as the prerequisite to persuade IBM to join OSF. Others suggest that IBM is making such a huge profit that it was eager to give over its operating system to OSF.

As with most issues, things are not so black and white. There is probably some truth to the assumption that OSF founders felt that choosing AIX would ensure that IBM would come on board. However, we don't think that it was a direct demand from IBM for participation. As to the money issue, we do not believe that the profit motive was the strongest in IBM's decision. IBM was simply following the example set by Sun Microsystems when it gave NFS to the world. Having AIX as the foundation for one of the two key standards-based operating systems provides IBM with instant credibility.

What's so special about AIX? Why is it that the Open Software Foundation was unwilling to use unaltered System V.4 as its operating system? Is AIX a much more advanced operating system than System V.4? What will users gain by using this operating system? In the complex world of operating systems, these questions cannot be answered in a single word or phrase. We will try to describe what we understand to be the most important characteristics that distinguish AIX from System V.4 and those features that are similar or even the same. Again, a caution. AIX is only the base operating system. It will be subject to the open process of OSF, which will ultimately decide which characteristics of AIX Version 3 will remain the same and which will be modified. The operating system that OSF offers later this year may have some significant differences from the one we are describing here.

The Kernel

The first—and, in this politically charged environment, perhaps the most important—characteristic is that OSF 1 will comply with AT&T's SVVD for System V.3. This does not mean that all the functionality of System V.3 will be implemented in AIX (which is based on System V.2), but there will be enough characteristics to pass AT&T test suites. In addition, like System V.4, AIX has integrated a majority of the BSD system calls.

One of the key differences between System V.4 and AIX is that AIX will have a paged kernel. In a paged kernel, modules are brought into the kernel at run-time, similar to the way a device driver might be brought in and out of the kernel in traditional Unix. However, this is not typical for other kernel functions. System V, in contrast, pages data in and out of the kernel. The benefit? It provides for a more efficient use of physical memory so that application designers don't have to worry about the amount of space tables take up.

Virtual Memory Implementation

The Virtual Resource Manager (VRM) had been a distinct entity in Version 2 of AIX. The original VRM work was separate for pragmatic reasons. IBM used Interactive Systems to do the original work on Version 2. Therefore, it was easier to bundle the enhancements and hand them over to Interactive. But having virtual memory outside the kernel slowed down the operating system enormously. Therefore, when IBM began designing Version 3, developers made the VRM part of the kernel, both for performance reasons and to allow for easier portability. In this area, OSF will have major work to do, because, at present, the VRM is still closely tied to the IBM PC/RT hardware.

Virtual memory in AIX includes three parts:

- A logical volume manager, which manages disk space at the logical level and controls the disk-mirroring (of up to two copies). It is an extensible group of disk partitions that divide the physical disk into equal 4MB partitions. These partitions can be dynamically expanded.

- A Virtual Memory Manager (VMM), which supports the RT file system. The VMM allows for the distributed execution of programs at remote sites.

- A simplified file system, Distributed Services (DS), whose complexity has been reduced through the use of a virtual memory manager that was initially designed for the RT. DS allows for distributed executes so that remote sites can execute a program from a server.

The Virtual Memory Manager assumes that there is a logical volume manager below it through which all storage can be viewed or accessed. The key to the implementation's power is that the VMM doesn't have to deal with bad blocks or mirroring. The Logical Memory Manager handles these complications. This allows the VMM to see a perfect volume and, in the end, simplifies coding.

The new implementation of the VRM is based on the concept of a pageable kernel. (Both System V and BSD page data in and out of the kernel, but their kernels are not pageable.) AIX is the only Unix kernel that has a paged kernel. In addition, user processes are also paged. The user does not copy information into an address space. Instead, modules are dynamically loaded at run-time. Dynamic loading is implemented in System V.4, too, but information is dynamically loaded into user space rather than into the kernel.

Like System V.4, OSF 1 implements single-level store to eliminate the use of buffer cache.

File Systems

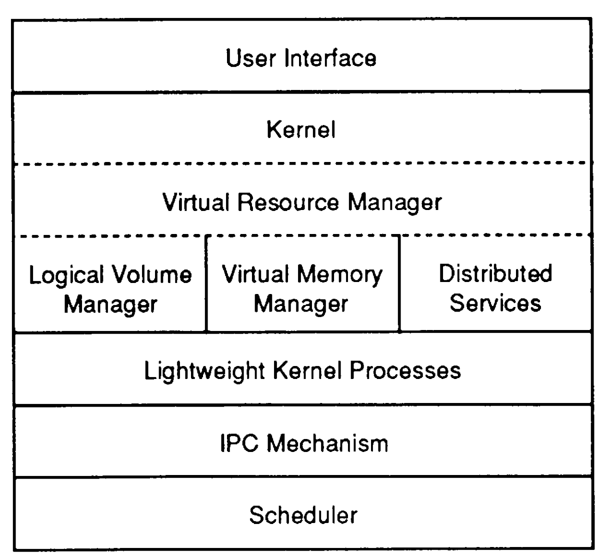
OSF 1's method for handling file system technology is still unclear. OSF's initial plans were to adopt IBM's Distributed Services (DS), a facility similar to Sun's NFS. DS technology is, in some aspects, superior to NFS. For example, DS allows files to be paged across the network, offers statefulness, and does not require something like Sun's Yellow Pages for reliability. With DS, users can build logical volumes that can span multiple disk volumes. The system administrator can allow these logical volumes to grow.

Given the charged atmosphere surrounding compatibility, OSF will probably begin by adopting NFS because it is a de facto industry standard. However, as it does in System V.4, OSF 1 will use Sun's vNode switch to allow users to move between different file system implementations.

Reliability

OSF 1 will include some important commercial reliability characteristics. For example, it will use database journaling technology and atomic commits. Both these features are unique to AIX and enable recovery in case of a system crash. While the degree of robustness of the file system is comparable to System V.4, file system recovery is more efficient and faster under OSF 1.

An Early View of the OSF 1 Design



One key advantage of AIX is disk-mirroring support, which allows users to write the same data to multiple disks simultaneously. Disk-mirroring is done dynamically. Although it has been implemented in fault-tolerant versions of Unix, disk-mirroring has never been available in a mainstream version of Unix.

Networking and Communications

AIX has implemented a distributed computing facility similar to Streams. This involves creating lightweight kernel processes. In AIX Version 3, users will be able to create kernel processes that have no associated user space. They are used to implement network protocols in the kernel and process traffic in the network.

However, given the wide acceptance and elegance of Streams, we expect that OSF may indeed decide to offer it as an alternative to lightweight kernel processes.

Real-Time

Both System V.4 and OSF 1 will improve real-time processing. Both operating systems have made the kernel preemptive. However, there are some subtle differences. System V.4 requires that a kernel process finish before a real-time process can begin. Therefore, real-time response could suffer. In OSF 1, a context switch can be set up at any time so that a real-time process can instantly be activated. This allows for absolute processes to be established. The real-time priority of an absolute process does not decay over time. This is made possible by a new IPC mechanism that supports a new form of sleep and wake. It includes a wake signal that alerts the scheduler that events have occurred and forces current processing to be interrupted.

Multiprocessor Support

Part of the implementation of preemption in OSF 1 has restructured system I/O so that it can potentially support thousands of processes. This is a key component of implementing multiprocessor support. Another key requirement is the preemptable kernel, which, as mentioned above, is part of this first version of OSF 1. What, then, is left to do? Most of the remaining work has to do with developing mechanisms for porting signals and interrupts from one processor to another. OSF has the option of integrating various implementations of multiprocessing into its operating system. For example, Honeywell Bull has offered its implementation of multiprocessor support to OSF. One of the biggest problems of implementing multiprocessing under Unix is the overhead of locking and unlocking processes in the kernel.

Scheduler

OSF 1 has changed the System V scheduler to make it preemptive and intends to move to make it class based.

National Language Support

Both System V.4 and OSF should have a comparable level of National Language Support (NLS). Neither operating system has begun to implement double-byte characters. However, Hewlett-Packard has offered this technology to OSF, so it will probably make its way into the operating system. We expect that member companies will express their urgent need for comprehensive NLS, and it will be a key priority for OSF's next version.

Shared Libraries and Memory

Like System V.4, OSF 1 will support shared libraries and memory. OSF intends to provide generalized dynamic loading for both the kernel and user space. Shared libraries can be loaded on demand.

User Interface: Motif

The user interface selected by OSF will be called "Motif." It is based on a combination of several technologies offered during the RFT (Request For Technology) process. The most important part of the new user interface is that it will maintain consistency with the PC environment by adopting the look and feel of Presentation Manager on top of X-Window (X.11.3). This technology was taken from a proposal made jointly by Microsoft and Hewlett-Packard (HP). The key elements included from HP are:

- The HP 3-D appearance
- Window manager
- Style Guide
- National Language Support
- HP's API

The underpinning for Motif comes from DECwindows. The following parts of DECwindows will be part of Motif:

- Interface toolkit.
- Applications Programming Interface (API).
- Digital's User Interface Language (UIL) compiler and resource manager. The UIL helps users make changes in the presentation of information within windows.
- An icon grouping that helps users find items not displayed on the screen.

Motif will not try to provide all the details of how the desktop will actually look. It will, however, have a style for where functions should be placed on the screen and how features such as pull-down menus will be implemented. OSF wants to leave enough room for each vendor to add value at some level.

OSF has also set up two other technology categories in addition to the selected user interface pieces. First, there is the Technology Catalogue. This is a group of three interesting technologies that, once perfected to meet the needs of OSF members, could become part of Motif. These include Apollo's Open Dialog presentation and dialogue manager; Base/OPEN, a user interface management system submitted by Swedish Telecom Group; and Generic Window Manager, a user interface management system submitted by Groupe Bull. OSF has also established a research program for technologies that are under consideration for future integration into the user interface. The first technology to be considered is Carnegie-Mellon's Andrew Toolkit.

No decision has yet been made about display languages. For example, PostScript is not part of Motif because of pricing factors, but individual vendors could choose to add PostScript to their versions.

Licensing terms are key to this user interface selection. From OSF's perspective, the company needed to bring out a user interface technology that would be priced low enough to encourage widespread use. The way to capture marketshare may well be through the interface, a concept that OSF understands well. Therefore, licensing is inexpensive. A source license will cost $1,000 per copy; a binary license will cost between $16 and $40, depending on size of machine; a run-time license will be priced as low as $4 and as high as $10. Universities can purchase a site license for only $1,000.

Licensing of OSF 1

OSF has gotten off to a good start with its pricing for user interface, but pricing the operating system itself may not be so simple. Each vendor selecting OSF 1 will first have to purchase an AT&T license, so it seems obvious that OSF 1 will have to cost more than System V.4. However, if Unix International puts enough pressure on AT&T to lower licensing costs, the difference may not be as significant.

Security

Security will become an increasingly important requirement. OSF 1 will implement DOD C2 level security. However, OSF developers are under pressure to exceed this level and will probably move to B1, a higher level of security.

System Administration

Many of the administrative functions within AIX were taken from System V and from BSD. There is one exciting change that will have ramifications for the future. AIX includes an object data manager that replaces a lot of text files with an object-oriented database. (In traditional Unix, for example, the user list and the system configuration files are simple ASCII text files.) With object-oriented database technology, each object could include historical and behavioral characteristics in a field. For example, a user name could include information about the history of that user, including which facilities he or she was allowed to access under which conditions. While object-oriented databases are not new, the application of this technology to a production operating system is.

Conclusions

This snapshot of two emerging operating systems shows clearly that we are at the beginning of some important changes in the way Unix is designed and implemented. Much work remains to be done. Multiprocessing needs to be added to both kernels, and security needs to be enhanced. The good news is that these are incremental changes. The foundation for the future look of Unix has been set with the development of System V.4 and OSF 1. These operating systems will be the base for the next three years.

At the same time, we expect that operating system technology will continue to progress. We anticipate that Sun Microsystems will continue its work to rewrite the Kernel with C++ so that it becomes modular and, therefore, easier to enhance. We expect that OSF will also look at ways of moving to an object-oriented kernel as well. OSF has established a Research Institute to begin exploring the possibility of incorporating research technology into future generations of systems and software. Unix International will set up a similar organization.

When Sun's developers are successful in bringing out a new object-oriented kernel, we imagine that Unix International and its partner, AT&T, could begin to merge it into the next version of System V. At the same time, OSF would competitively work to ensure technological equivalence.

Despite these interesting changes on the horizon, we hope that discussions over the next few years will move beyond the kernel. If Unix is to truly become a commercial platform, it should be much more important to discuss issues such as user interface, applications design, and training. ◐

This article is part of a special report on OSF and AT&T/Unix International that will be available in the spring.

Announcing

PATRICIA SEYBOLD'S THIRD ANNUAL TECHNOLOGY FORUM

TITLE

OBJECT ORIENTATION

Defining the end-user platform for the '90s

DATES

April 3,4, 5, 1989

LOCATION

Boston Marriott/Cambridge
Cambridge, Mass.

PRICE

- $895

Discounts for two or more attendees from the same company are available. For more information, call Deborah Hay at 1-800-826-2424.

SPONSOR

Patricia Seybold's
Office Computing Group

"This series [OCG Forums] is probably the best prepared and managed of any I've attended in 20 years. Well done!"

Jack Russ
Lawrence Livermore
National Laboratory

1988
Seybold Executive Forum Attendee

OVERVIEW

The 1989 Technology Forum will define and explore the different manifestations of the major architectural platform for the next decade: Object Orientation. Attendees should obtain not only a basic working understanding of the various aspects of object orientation, but of implementation issues and trends as well.

TOPICS

We are dividing the very broad subject of object orientation into three primary areas: Platforms, Implementation Issues, and Applications.

Platform Sessions will examine the operating systems and database software foundations for true object orientation. We will also explore the emerging compound document architecture standards as mechanisms for the integration of heterogeneous objects.

Implementation Issues Sessions will examine the evolution of database managment systems toward object orientation and issues in interchange of compound documents.

Applications Sessions will highlight end-user experiences as well as look into the future of academic and research applications.

Companies presenting include:

Data General Corp.
Christopher M. Stone, *Group Manager, Work Group Software*

Graphael
Joe Hancock, *President*

Hewlett-Packard Company
Bill Crow, *NewWave Research & Development Manager*

Houston Chronicle
Jack Stanley, *V.P., Operations*

Imnet Corporation
David B. Black,
V.P., Research & Development

Index Technology
Burt L. Rubenstein,
V.P., Chief Technologist

Metaphor Computer Systems, Inc.
E. David Liddle, *Chairman*

Neuron Data
Patrick Perez, *CEO*

Object Design, Inc.
Thomas A. Atwood,
Chairman

What Past Attendees Have Said About Our Forums:

Patricia Seybold as a seminar leader: "10" (highest rating). "Very professionally organized and carried out!"
John Bratlie, Boeing

"This was the most impressive array of speakers, all focusing on a common theme, I've ever seen. Brilliant minds!"
Jean Straehl,
Com/Energy Services Co.

"Superbly planned, managed, and executed!"
Tom Karol,
Analog Devices, Inc.

"Worth every penny."
Lauri Hirvonen,
Nokia Data Systems, Finland

Who Should Attend?

The Technology Forum presents in-depth information on computing trends and emerging systems and software. Such information is critical to the planning needs of:

- Corporate Technologists
- Application Developers
- MIS and DP Professionals
- Product Planners for Computer Vendor Corporations

Companies presenting include:

The Office Computing Group
Michael D. Millikin,
V.P., Chief Technologist

Oracle Corporation
Larry Stevens, *Director, Tools & Marketing, Product Division*

Plexus Computers, Inc.
Tom Frederick, *V.P., Sales & Marketing*

Proteo Technology Corp.
Roger Tessier, *Chief Architect*

Relational Technology, Inc.
Carol Realini, *Director of Product Development*

Servio Logic Corporation
J. Michael Connell,
General Manager,
Data Management Systems
David A. Taylor, Ph.D.,
Business Unit Manager,
Business Applications Group

Soft•Switch, Inc.
Michael D. Zisman, *President*

Sybase
Dr. Robert Epstein, *Exec. V.P.*

USE THE ORDER FORM BELOW OR: **CALL TOLL-FREE 1-800-826-2424!**

☐ **PLEASE REGISTER ME FOR PATRICIA SEYBOLD'S TECHNOLOGY FORUM!**

$895 per person, group discounts are available.

Send registration form and payment to:
Patricia Seybold's Office Computing Group
148 State Street, Suite 612
Boston, MA 02109
Fax: 617-742-1028

Cancellation Policy:
Should a registrant be unable to attend the Forum, The Forum Office will refund the full registration if notified before March 1. Cancellations from March 1 to March 15 are subject to a $75 service charge. There will be no refunds as of March 16. Substitutions may be made at any time.

Name ____________
Title ____________
Company ____________
Address ____________
City ________ State ____ Zip ______
Telephone ____________

☐ Check Enclosed (please make checks payable to Patricia Seybold's Office Computing Group)

☐ Charge to my ☐ MasterCard ☐ VISA ☐ American Express
Card No. ________ Exp. Date ________
Signature ____________
☐ My Purchase Order Number: ________

•TUTORIAL•

Extending Your User Interface

Even nonprogrammers can build complex tools from simple ones.

By Gary J. Nutt

In our last column, we discussed how the Unix shell can be customized to fit one's individual needs and tastes. A little bit of this knowledge is mandatory to provide a workable interface, and each user is expected to set at least some of the parameters in order for the shell to be usable at all.

The shell also provides a mechanism by which one can build complex tools from simpler ones. It is even possible to construct your own pieces to incorporate into a composite command, although that is not necessary for many tasks. Of course, this latter capability assumes that you are willing to write some C programs; however, nonprogrammers can also take advantage of the ability to glue existing tools together to accomplish their own purposes.

What are the basic tools with which we have to work? First, there is a collection of different shell commands that read input from "stdin" and write results to "stdout". Each such command performs some operation on the input, yielding an output, so these shell commands are called filters. That is, a filter is a device into which we "pour" input and out of which we expect to get some altered output. For example, we can pass an ASCII file into the grep filter (command), and it will filter out all lines that do not contain at least one instance of some string. So, if we have a file named "addresses" containing groups of lines such as:

Patricia Seybold
148 State Street, Suite 612
Boston, Massachusetts 02109

Gary Nutt
3450 22nd Street
Boulder, Colorado 80304

then we can use the grep filter

grep Massachusetts addresses

to remove all lines that do not contain the string "Massachusetts", thus printing the line

Boston, Massachusetts 02109

on stdout (the terminal). That is, grep filtered out all lines that did not contain the string "Massachusetts" and printed all the remaining lines. Chapter 1 of your Unix User's Manual lists the set of filters provided with your version of Unix (they are different for individual versions of Unix).

In addition to basic tools, the shell provides "glue" for interconnecting the tools: The simplest tool is I/O redirection—the "<" and ">" symbols that can appear before filenames on a command line. The "<" symbol means that the filename that appears to the right of the symbol is to be substituted for stdin when the command is executed. Thus, we could have typed

grep Massachusetts <addresses

in place of the line above, and grep would produce the same result. (In this case, grep is "smart enough" to interpret the second parameter as a filename to substitute for stdin. Thus, we could write "grep Massachusetts" as a command line if we wanted grep to search stdin—the characters that we are typing at the keyboard.) The ">" symbol is a mechanism for redirecting the output from the filter to a named file instead of to stdout. Thus, the command line

grep Massachusetts >list <addresses

would cause the line

Boston, Massachusetts 02109

to be written into a file named list. So, the I/O redirection symbols are a means by which we can make use of files for the source or destination of any shell command.

We have also previously described pipes as a means for combining shell commands. A pipe is designated by the "|" symbol on the shell command line. The effect is to redirect the output of the command on the left side of the "|" into the input of the command on the right side of the symbol. A more subtle implication of the use of pipes is that, if one types

abra | cadabra

to the shell, it will actually start two commands (abra and cadabra) executing in parallel, with the stdout of abra being the same as the stdin of cadabra. Conceptually, it is if we had typed the two shell lines

abra >temp&
cadabra <temp

where the "&" symbol is the Unix fork command described in the previous columns. (The distinction between the two is that the pipe method doesn't really have a file named "temp" declared—instead, the stdout of abra is simply routed directly to the stdin of cadabra.)

Let's look at a real example using pipes: The wc command will read a file from its input, then print the number of lines, words, and characters followed by the file name onto the output. So, if we type

grep Massachusetts addresses | wc

we can expect that grep will produce one line of output (see above) and that wc will read this line as input; it will then print the following line on the display:

1 3 28

(The input filename is omitted by wc if the input file is piped into the command.) That is, wc sees one line with three words made up of 28 characters. If we typed

grep Street addresses | wc

the composite command would print

2 8 45

since there are two lines that contain the string "Street", namely

148 State Street, Suite 612

and

3450 22nd Street

Pipes and I/O redirection are the fundamental glue for combining shell commands. Let's look at some other ways we can use them to build up more complex tools, i.e., command lines that have multiple commands connected with the pipe symbol.

Suppose that we chose to lay out the address file differently, putting all of the information about one entry onto a single line (even if the line is too long to fit on one line of the display). We will use a semicolon to separate parts of the record, so that we can look at a line and read the name up to the first semicolon, the street address up to the second semicolon, etc.

George Bush; 1600 Pennsylvania Ave.;
Washington, D.C. 00001; (212)555-1234
Larry Bird; The Boston Garden;
Boston, Massachusetts 02110; (617)555-3333
Mickey Hatcher; 34 Chavez Ravine;
Los Angeles, California 90423; (213)555-3434
Magic Johnson; The Forum;
Englewood, California 90300; (213)555-3333
Mickey Mouse; 123 Magic Lane;
Fantasyland, California 90123; (213)555-6789
Gary Nutt; 3450 22nd Street;
Boulder, CO 80304; (303)444-0341
Patricia Seybold; 148 State Street, Suite 612;
Boston, Massachusetts 02109; (617)742-1028

Now, if we want to know Larry Bird's phone number, we type

grep Larry addresses

resulting in

Larry Bird; The Boston Garden; Boston, Massachusetts 02110; (617) 555-3333

being printed on the display. We have constructed an online address book with a search feature simply by formatting an ASCII file to contain a record per line and by using grep to search the file. Of course, there are some warts on our little "system," since, if we type,

grep Mickey addresses

we will get

Mickey Hatcher; 34 Chavez Ravine;
Los Angeles, California 90423; (213)555-3434
Mickey Mouse; 123 Treadmill Lane;
Fantasyland, California 90123; (617)555-6789

Or, if we type

grep Magic addresses

we will get

Magic Johnson; The Forum;
Englewood, California 90300; (213)555-3333
Mickey Mouse; 123 Magic Lane;
Fantasyland, California 90123; (617)555-6789

We can search our address file using any arbitrary criteria. For example, we could find all lines that contain the string "Boston", which will ordinarily give us a listing of all entries that live in Boston (unless, of course, Ralph Boston's address and phone number were in our address file). Generalizing slightly, the line

grep Boston addresses | wc

would print a line that indicated—among other things—how many lines in the address file contained the string "Boston". We could easily query our address file for other such interesting information, for example, typing

grep "(213)" addresses

to list the names of people that are in the 213 area code.

Suppose that I wanted to make a mailing list to all entries in the address file that lived in the 213 area code. Then the previous command would print

Mickey Hatcher; 34 Chavez Ravine;
Los Angeles, California 90423; (213)555-3434
Magic Johnson; The Forum;
Englewood, California 90300; (213)555-3333
Mickey Mouse; 123 Magic Lane;
Fantasyland, California 90123; (213)555-6789

This could be redirected to a file by redirecting the output, for example, by typing

grep "(213)" addresses >LAaddresses

Unfortunately, the entries in the list are not in a form suitable for printing on address labels. While it would be nice if there were some shell command that could reformat a line such as

Mickey Hatcher; 34 Chavez Ravine;
Los Angeles, California 90423; (213)555-3434

into a line such as

Mickey Hatcher
34 Chavez Ravine
Los Angeles, California 90423

we may choose to write a C program to do this (actually, we could also use "sed" to accomplish the task). The C program would be simple: It would read characters from stdin until it saw a semicolon, then it would emit the characters terminated by a newline character in place of the semicolon. It would do this three times for each input line, then skip the last field (the telephone number), printing a blank line in its place. Suppose the program had been written, compiled, and stored in /usr/gjn/bin/mailingaddress. Now, we can type

grep "(213)" addresses | mailingaddress >mailinglist

which will create a file named "mailinglist" that contains

Mickey Hatcher
34 Chavez Ravine
Los Angeles, California 90423

Magic Johnson
The Forum
Englewood, California 90300

Mickey Mouse
123 Magic Lane
Fantasyland, California 90123

Just as with many other aspects of Unix, this one is easy to forget; if the complex command is not used very often, then we may have to reinvent the same command line each time we want to use it. In order to get around this problem, it is possible to simply store the command line into a shell script file named, say, "printaddresses". Now, we could simply type

printaddresses

each time we wanted to generate an address list for area code 213. (We must be careful that printaddresses has "execute" permission—see the "chmod" command in the manual.)

If we wanted to print an address list for area code 617, we could edit the file to change the 213 to a 617 and then invoke the shell script. However, it is also possible to write shell scripts with dummy arguments in them instead of real parameters. Suppose that we wrote the line in printaddresses to be

```
grep $1 addresses | mailingaddress >mailinglist
```

where the "$1" replaces the string search argument for the grep command. Now, we can type

```
printaddresses "(213)"
```

to print the addresses in area code 213, and

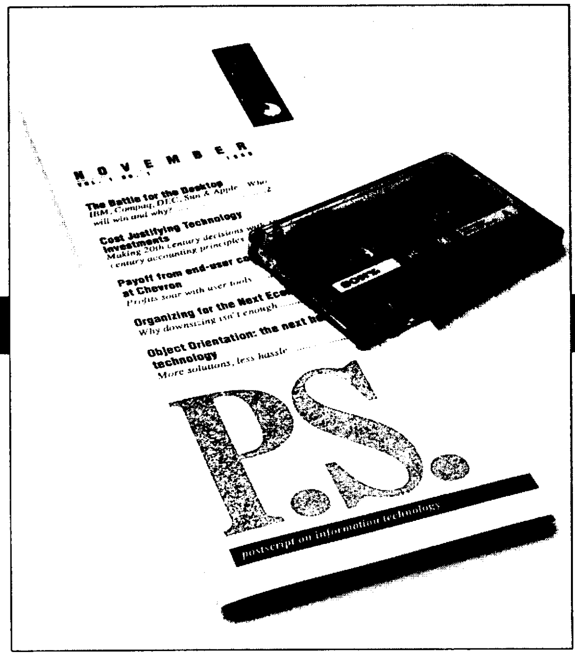
```
printaddresses "(617)"
```

for area code 617. That is, the first parameter on the shell script's command line is substituted for "$1" in the shell script before it is executed.

It is possible for a shell script to become very sophisticated—after all, it is merely a high-level programming language for the computer. However, the language has very simple rules for composing programs from other programs. Thus, it is possible to program the shell with minimal time investment into the language; the programs are barely more sophisticated than stored keystrokes (we have only added I/0 redirection and pipes). The shell user interface is based on one of Alan Kay's basic notions about user interfaces to computers: Simple things should be easy to do, and complex ones should be possible. ◐

P.S.

postscript on information technology



Patricia Seybold Launches a New AudioNewsletter

CURRENT TRENDS IN NONTECHNICAL TERMS

This innovative monthly service, both an audiocassette and a printed newsletter, is designed for you, the technology watcher, and for your nontechnical colleagues. It comes to you from Patricia B. Seybold, publisher of *The Office Computing Report, UNIX in the Office*, and *Network Monitor*. Every issue presents discussions of current issues and technology trends in nontechnical terms so that you and your associates can find out what is going on and make more intelligent decisions.

FASCINATING ONE-ON-ONE INTERVIEWS

Each month, Patricia Seybold will bring you fascinating interviews on a variety of topics, including:

- Executive Information Systems
- Cost Justification
- Object Orientation
- Evolving Standards
- Desktop Platforms
- End-User Computing
- Managing Change
- Partnering for Progress

She will also give you her personal view of where things are headed. . .what to watch. . .and what you can safely ignore.

LATEST LEARNING AND READING TECHNOLOGY

The innovation doesn't stop with our new audio format; the printed newsletter is designed according to the latest research in learning and reading technology. You can skim the whole newsletter in less than one minute, easily locating and assimilating the points you want to pursue.

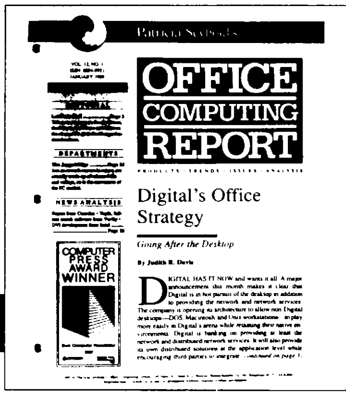
INNOVATIVE AUDIOTAPE REVEALS EVEN MORE

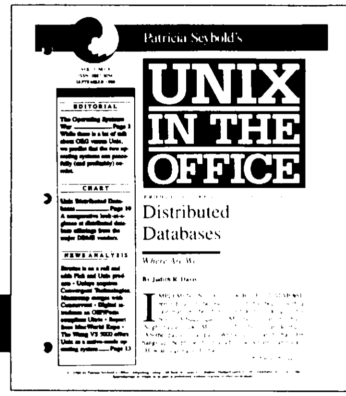
But that's just the beginning. . .because an exclusive audiotape that accompanies each printed edition of the newsletter reveals even more:

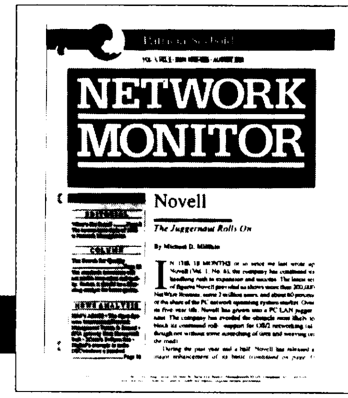
- One-on-one interviews with industry insiders
- Technology trends explained in plain English for busy executives who must make fast, intelligent decisions
- Firsthand success stories—how your peers manage change in their corporate cultures

P.S.

Listen to it in your car on your way to work or plug into it as you exercise!







OFFICE COMPUTING REPORT

BEST COMPUTER NEWSLETTER 1987

The Office Computing Report is the oldest and largest-circulation research report on office technology. Each month's feature article focuses on an important technology, architecture, information system, or software product. Additional coverage in-cludes user perspectives, issues, and implementations.

The Office Computing Report was named **Best Computer Newsletter** at the 1987 Computer Press Awards.

RECENT FEATURE ARTICLES

Document Processing
We defined the coming generation of document processors, which will include word processing and desktop publishing functionality, object orientation, and a graphical user interface.

Metaphor Computer Systems
We took an in-depth look at this innovative object-oriented, end-user system.

Digital's Office Strategy
We evaluated Digital Equipment's newly-announced strategy for winning the battle for the desktop.

UNIX IN THE OFFICE

UNIX IN THE COMMERCIAL WORLD

UNIX in the Office is a monthly publication focusing on applications and technology trends in the Unix arena. Each monthly issue spotlights a major vendor, strategy, technology, or industry trend. In addition, feature articles frequently cover Unix databases, operating systems, innovative applications, and standards issues.

UNIX in the Office was launched in February, 1986.

RECENT FEATURE ARTICLES

SmartWare from Informix Software
As one of our in-depth product analyses, we positioned SmartWare in the market of rapidly growing office applications.

IBM's Unix
We assessed IBM's new vision of becoming a premier Unix system vendor with a large installed base.

AI and Unix
We spelled out just how AI's decision analysis and support systems are helping managers to solve common business problems.

NETWORK MONITOR

LINK UP WITH THE #1 LAN REPORT

Network Monitor is a monthly research report that provides comprehensive analysis and objective evaluation of LAN products and vendor strategies. The report assesses product strengths and weaknesses in terms of on-the-job performance and user concerns.

In addition, regular departments cover connectivity and integration, and incisive news analysis probes the "why" as well as the "what."

RECENT FEATURE ARTICLES

3Com
We discovered how 3Com stepped from one window to the next without suffering wrenching financial and/or product line transitions.

OS/2 Networking
We examined the emergence of OS/2 and its relationship of client to server from the viewpoints of IBM, Microsoft, 3Com, and Novell.

OSI Standards
We took a pragmatic perspective on OSI. How will vendors migrate from existing protocols to OSI protocols?

NEWS ANALYSIS

PRODUCTS • TRENDS • ISSUES • ANALYSIS

• INSIDE •

•DIGITAL•

VMS to be Posix Compliant

On January 10, Digital presented a comprehensive set of program and product announcements that focus on the desktop as a key element of a completely integrated computing environment. The company has taken its traditional "single system" approach and molded it into a strategy that makes sense for the 1990s. Included is a strong commitment to Digital's first coherent desktop strategy, as well as a continuing commitment to distributed network computing.

POSIX COMPLIANCE. One of the most significant strategy announcements is Digital's intent to make its VMS operating system Posix compliant. (This was accomplished for Ultrix with the introduction of Ultrix-32 last year.) The short run advantage, from a marketing perspective, is that DEC will be able to sell VMS as a standards-compliant operating system. In the long run, it will allow DEC to maintain a robust proprietary operating environment and still comply with and compete in the standards arena.

Not only will DEC's two operating platforms have a consistent user interface in DECwindows, but, because DEC is writing all software in the C language, it will be able to easily port software to both environments. Posix compliance will not be a layered product on top of VMS but rather part of the operating system kernel, thus providing a high-performance Posix interface.

New RISC Platform

One major development here is the introduction of a powerful new RISC-based workstation. Thus, the hardware component of Digital's architecture is evolving to include both VAX- and RISC-based platforms. DEC also expanded its desktop offerings with new VAXstations and the company's first industry-standard PCs. (Note: In Digital's parlance, Digital workstations based on the VAX architecture are VAXstations; all other workstations are DECstations.)

RISC WORKSTATION. The DECstation 3100, code-named the P-MAX, is based on a RISC (reduced instruction set computer) chipset developed by MIPS Computer Systems, Inc., and it runs only Ultrix-32 (Version 3.0). Digital maintains that its "Sun killer" is four times faster than the Sun 3/60 or Apollo DN3500 and twice as fast as the Sun 4 or Apollo DN4500, yet is priced much lower than all of these competitors. The DECstation 3100 performs at 14 MIPS, based on the Dhrystone benchmark. It includes an 8-plane 15" or 19" monitor with 1,024 x 864 pixel resolution, 8 to 24MB of memory, a mouse, an SCSI port, Ultrix-32 license, and support for TCP/IP and NFS. Digital's aggressive pricing starts at $11,900 for a base-level, 15" monochrome workstation with 8MB of memory. The 3100S, a server version with 24MB memory and 996MB of disk storage (and no graphics subsystem or monitor), starts at $43,400. The DECstation 3100 is available now. Shortly, Digital will also introduce a C compiler to enable the porting of VAX-based Ultrix applications to the new RISC platform.

Digital has delivered a very aggressive price/performance scenario with its RISC workstation, which is the reason Digital went to RISC in the first place. In the future, we expect to see technical- and engineering-class workstations priced cheaply enough to sit on desktops.

NEW VAXSTATIONS. The VAXstation 3100, the widely-anticipated P-VAX or personal VAX, is based on the same CMOS processor as its less powerful VAXstation siblings. However, it provides three times the performance of the VAXstation 2000. It features a new compact desktop configuration to reduce the footprint on the desk, an SCSI port for the addition of standard peripherals, a 15" or 19" monitor with 1,024 x 864 pixel resolution, 8 to 32MB of memory, a mouse, up to three 3.5" and two 5.25" disk drives, and either VMS/ Desktop-VMS or Ultrix as the operating system. Digital's new RRD40 CDROM drive (600MB) is an option starting at $1350. A basic diskless system priced at $7,950 includes 8MB of memory and a 19" monitor. A server model starts at $38,000. These will begin shipping in March.

The VAXstation 3520 and 3540 are high-end VAXstations designed for compute-intensive, high-performance 2D and 3D graphics applications. According to Digital, they are the industry's first workstations to bring high-resolution 2D/3D graphics to the X-Window environment. They are symmetric multiprocessing workstations with two or four CMOS processors, a new high-resolution 19" monitor (1280 x 1024 pixels), and an optional 24-plane display for true color. A key feature is support for PEX (PHIGS extension to X-Window), making it possible for applications to be ported to run on other platforms that support the X-11 Window System Version 11. Memory ranges from 8 to 64MB, and storage can reach 1.3GB. Digital stated that availability and pricing of the 3520 and 3540 will be announced later this quarter. ◐ —*J. Davis*

•ORACLE•

A New Office Contender

Oracle Corporation, best known for its relational database management system (RDBMS), has formed a new Office Automation Division and released its first OA product, Oracle*Mail. Based on the Oracle RDBMS, Oracle*Mail is an enterprise-wide office communication system. The company's goal is to provide transparent communication of all forms of information throughout an organization. Oracle*Mail is designed to be a general purpose communications facility, providing not only user-to-user messages, but also communications between users and applications and between applications. The company plans to port Oracle*Mail across all of the platforms and networks that Oracle supports, thus providing the underpinning and communications backbone for future OA modules from Oracle.

This is Oracle's second foray into the applications arena, following the company's announcement of Oracle Financials, a suite of accounting application software packages also based on the Oracle RDBMS. Oracle Financials was the first step in the company's strategy to move beyond application development tools and to offer the customer full business solutions. The formation of the OA group is another facet of this strategy. The Oracle OA products are aimed directly at competing office systems platforms, such as Digital's All-In-1. While we think the company is underestimating the ability and commitment of these traditional office vendors to update and adapt their products, it will be interesting to see how successful a newcomer such as Oracle is in this marketplace.

MULTIPLE PLATFORMS. One of the advantages Oracle has over traditional office systems is the fact that its underlying RDBMS runs across heterogeneous platforms, operating systems, and networks. This level of portability and flexibility will appeal to organizations faced with multivendor environments. Oracle also intends to provide independence at the Presentation Manager level in its OA products. This preserves the native interface environment for the user, regardless of what system is on the desktop—X-Window, character-mode, block-mode, etc.

INTEGRATED COMMUNICATIONS. Oracle intends to make Oracle*Mail an open product and has built in multiple levels of access for different users and applications. A command line utility provides an interface for the non-programmer to the mail system (e.g., to generate a report with a report writer and send it through the mail system to a distribution list). Callable routines provide a programmatic interface to integrate CASE, application tools, utilities, application programs, etc. with Oracle*Mail. Oracle will also publish the full API for Oracle*Mail this spring, allowing complete open access for those customers or developers who want to do sophisticated integrated applications.

Other plans include embedding a closer relationship with Oracle*Mail in future releases of Oracle's current application development tools, such as SQL*Report, SQL*Forms, and Oracle Financials.

EASE OF USE. Oracle*Mail uses ring and pull-down menus, accelerator keys, and lists of values from which the user can select. The full-screen editor provides basic editing of mail messages, including word and line editing functions, word wrap, and cut and paste (within and between mail messages). The user can also specify another word processor as the default mail editor (e.g., WPS-Plus on Digital's VAX). The on-line, context-sensitive help is fully cross-referenced with an index and a table of contents.

The user can create a message template for any type of frequently generated message, such as a phone message or meeting notification. Another interesting use of this feature is to generate and communicate company or system-wide forms.

Directory Services. The mail directory is distributed in Oracle*Mail—that is, a directory resides on each node. The administrator decides how extensive the directory is at each node—whether it contains only local users, all users, or something in between. For replicated directory information (for example, if a directory of all users is kept on every node), Oracle*Mail automatically updates each directory in the network. The directory is available on-line, and users don't have to know the host name for remote addressees.

DATABASE FEATURES. In addition to standard E-mail features, users can also manipulate mail messages as a personal database. Oracle*Mail provides both sorting and query options. The user can query any of the structured data associated with a message (e.g., sender, date, subject) using a query-by-example approach. Oracle is evaluating the use of full-text search capabilities on the contents of mail messages for a future release.

Another emerging requirement for E-mail systems is filtering, the ability to establish rules for the disposition of mail messages based on information such as the subject, the sender, or what type of mail message it is. Oracle is well aware of this need (one of the key developers of Oracle*Mail was involved in the Lens project at MIT) and states that a filtering capability is coming in a future release.

MINIMAL OPERATIONAL COSTS. The administrative function of Oracle*Mail, designed for simplicity and ease of use, is based on menus and forms. Since Oracle's underlying distributed database mechanisms make the network transparent to both the mail system and the DBMS, the administrator doesn't need to worry about what networks are involved (e.g., Oracle*Mail looks the same on DECnet as on TCP/IP). In addition, updates to replicated directory information are automatically made across the network.

Oracle*Mail stores only one copy of a message per server, regardless of the number of local recipients of the message. And as we would expect, it also only transmits one message copy from node to node. This reduces both network traffic and disk storage requirements. Oracle*Mail provides full store-and-forward capabilities as well as full backup and recovery.

OPEN ARCHITECTURE. Oracle is now developing a series of gateways to other mail systems so Oracle*Mail can plug into and play in an existing mail network. Current gateways to other mail products include VAX Mail and Unix Mail (the native mail environments on current Oracle*Mail platforms). As the company ports Oracle*Mail to other environments, it will also build gateways to the appropriate native mail systems. By mid-1989, the company expects to introduce an X.400 gateway as well as connections to facsimile and telex systems.

FUTURE OA MODULES. Oracle already provides its own spreadsheet processor, called SQL*Calc, and an interface to Lotus (Oracle for 1-2-3). Oracle plans to expand its OA applications to include word processing (probably an existing product from another vendor), calendaring and scheduling (most likely developed by Oracle), document management, and a distributed bulletin board.

PACKAGING AND AVAILABILITY. Oracle*Mail is available now for Digital's VAX/VMS and Sun's Unix, and it includes a limited purpose, run-time version of the Oracle RDBMS kernel. (Oracle runs wherever the Oracle*Mail server resides). A Unix System V version is coming shortly and the company plans to port Oracle*Mail across its Oracle platforms. (According to the company, only the front end of Oracle*Mail will run on DOS.) SQL*Net, Oracle's optional software for network support, is also required for each server in the network to provide network connections.

Oracle*Mail for the VAX ranges in cost from $2,000 to $70,000 and for Sun from $2,000 to $30,000. The mail gateways each cost 25 percent of the cost of Oracle*Mail. ◐ —*J. Davis*

•USER INTERFACE•

A Motif-Supported Shell

X.desktop from IXI Limited (Cambridge, England) is among the first products to support Motif, OSF's user interface. Actually X.desktop was a candidate for the OSF user interface. At the time of the request for technology (RFT), OSF was considering a graphic shell component for its user environment. IXI's X.desktop, a graphical desktop management program, made it to the final rounds before OSF decided against a graphical shell selection. In the meantime, IXI has tailored X.desktop to support both Motif and Open Look. The company is also submitting X.desktop for the OSF Catalogue Technology Program.

X.desktop provides the details of how the desktop actually looks, whereas Motif doesn't. (Motif provides a style guide, but opted to leave room for each vendor to add value in this area.) It's an application program which is always running during a user session. It gives you graphical, Mac-like access to Unix, keeping the operating system out of sight. Objects in the system, such as tools, programs, data files, and directories are displayed as icons. The program's windowing system lets you access various applications and operating systems.

IXI has been forward-thinking in its strategy for X.desktop. The product was developed with standards in mind. In addition to meeting the style requirements of both Motif and Open Look, it's Posix-compliant and is based on the X-Window system (Version 11 Release 2). It also conforms to X/Open specifications. Furthermore, IXI has given X.desktop a configurable interface, so it can be tailored to various environments. A number of companies, including Locus and Olivetti, have already licensed X.desktop and bundle it in their Unix offerings.

With X.desktop, IXI is hoping to convert users ordinarily intimidated by Unix, and we're all for a less intimidating Unix environment. Multitasking capabilities coupled with freedom from operating system complexities should better the chances of Unix as a corporate contender. ◐ —*L. Brown*

Patricia Seybold's Computer Industry Reports

ORDER FORM

Please start my subscription to:

		U.S.A.	Canada	Foreign
☐ *Patricia Seybold's Office Computing Report*	12 issues per year	$385	$397	$409
☐ *Patricia Seybold's UNIX in the Office*	12 issues per year	$495	$507	$519
☐ *Patricia Seybold's Network Monitor*	12 issues per year	$595	$607	$619

Please send me a sample issue of:

☐ *Office Computing Report* ☐ *UNIX in the Office* ☐ *Network Monitor*

☐ **My check for $________ is enclosed.** ☐ **Please bill me.** ☐ **Please charge my subscription to: Mastercard/Visa/American Express** (circle one)

Name: ____________ Title: ____________

Company Name: ____________ Dept.: ____________

Address: ____________

City, State, Zip code: ____________

Country: ____________ Bus. Tel. No.: ____________

Card #: ____________

Exp. Date: ____________

Signature: ____________

Checks from Canada and elsewhere outside the United States should be made payable in U.S. dollars. You may transfer funds directly to our bank: Shawmut Bank of Boston, State Street Branch, Boston, MA 02109, into the account of Patricia Seybold's Office Computing Group, account number 20-093-118-6. Please be sure to identify the name of the subscriber and nature of the order if funds are transferred bank-to-bank.

UF-89-UX

Send to: Patricia Seybold's Office Computing Group: 148 State Street, Boston MA 02109; FAX: 1-617-742-1028; MCI Mail: PSOCG
To order by phone: call (617) 742-5200

Topics covered in Patricia Seybold's Computer Industry Reports in 1987/1988:

Back issues are available, call (617) 742-5200 for more information.

Office Computing Report

1988—Volume 11

#	Date	Title
1	Jan.	Office Computing: 1988
2	Feb.	Mac Attack
3	Mar.	Electronic Data Interchange—The New Fontier of Electronic Mail
4	Apr.	PC LANs Revisited
5	May	Remission or Recovery—Will NBI's New Direction Bring Renewed Life?
6	June	Service And Support *Second Feature:* LAN Support
7	July	AI Update—Expert Systems Flourish While the AI Industry Disappears
8	Aug.	Metaphor Computer Systems—A Quiet Revolution
9	Sep.	Technology That Supports Meetings
10	Oct.	Document Processing
11	Nov.	Upper CASE Tools—Getting Users to Help Design Systems
12	Dec.	Executive Information Systems

1989—Volume 12

#	Date	Title
1	Jan.	Digital's Office Strategy—Going After the Desktop

UNIX in the Office

1988—Volume 3

#	Date	Title
1	Jan.	Beyond Unix—Creating the Standard Operating Environment *Second Feature:* X/Open Champions Open Market
2	Feb.	Progress
3	Mar.	Pyramid Technology
4	Apr.	IBM's Unix
5	May	AI and Unix
6	June	SmartWare from Informix Software—The Quiet Contender
7	July	Olivetti—Global Bridges on a Standard Foundation
8	Aug.	BBN's Slate System—Delivering Compound Documents
9	Sep.	Distributed Databases—Where Are We?
10	Oct.	DECwindows
11	Nov.	Data General's Unix—A New Beginnning
12	Dec.	Document Processing with Interleaf *Second Feature:* Apollo and Sun Service and Support

1989—Volume 4

#	Date	Title
1	Jan.	Unisys Ofis Ensemble—In Search of Cohesion

Network Monitor

1988—Volume 3

#	Date	Title
1	Jan.	Taking Stock, Looking Ahead *Second Feature:* Mac to PC: Making the Connection
2	Feb.	Excelan—Riding the Crest with TCP/IP *Second Feature:* Update on Banyan
3	Mar.	3Com—Opening a New Window
4	Apr.	OS/2 Networking
5	May	Hewlett-Packard—Riding with the Wind
6	June	Exploring Distributed Network Computing
7	July	Transparent File Systems—The Locus Model for DNC
8	Aug.	Novell—The Juggernaut Rolls On
9	Sep.	OSI Standards—A Promise Fullfilled?
10	Oct.	ISDN: We Still Don't Know
11	Nov.	Data General's PC Networking Strategy
12	Dec.	3+Open—3Com Makes Its Move

1989—Volume 4

#	Date	Title
1	Jan.	APPN—Peer-to-Peer Networking Arrives at IBM