UNIX – The Force Behind Personal Computing?

By Dr. Charles Sauer, Director of Product Planning, Dell Computer Corporation

"Personal computing systems" can be defined in a very broad sense, from 8088 and 6502 personal computing machines on up through multiprocessor workstations. This article describes some of the current trends in operating systems for these systems and draws some conclusions based on those trends.

Evolution Of PC System Hardware

Both of the most influential personal computer designs, the IBM PC and the Apple Macintosh, became influential with simple operating systems (simple relative to operating systems found on larger machines). This was partly a natural consequence of the simplicity of the hardware systems. The original PC had a minimum memory of 16KB of primary memory and an *architectural maximum* of 640KB. The original Mac had exactly 128KB. Both had provision for diskette drives only. Neither had networking hardware (except slow serial links). The PC's graphics capabilities were extremely limited. The Mac had relatively high resolution video but limited display area. The PC had a primitive eight-bit I/O bus and the Mac had no expansion bus. Most importantly, neither had memory management units - there was no provision for protection or virtual memory (P. Norton and R. Wilton).

Since the introduction of these machines in the early 80's, the hardware systems have become dramatically more advanced in both capacity and function. Typical PC compatible machines and Mac's have at least a megabyte of memory and at least a 20MB fixed disk. It is estimated that 50 percent of these machines are connected to local area networks, and many are also connected to mainframes by traditional terminal connections. Usable bit-mapped graphics are typical on PC's, and higher resolutions/large displays are common on both PC's and Mac's. **The Industry Standard Architecture** 16-bit bus of the PC/AT is being challenged by two 32-bit buses, the **Extended Industry Standard Architecture** and the IBM Micro-Channel. The Mac II family supports the 32-bit NuBus. Typical PC's have either a 286, with segmented virtual memory support, or a 386/486 with both paged and segmented virtual memory support. Mac II's have either a 68020 with a 68851 option for paging or a 68030 with built-in paged virtual memory. Cached memory architectures are standard on high-end PC's and the latest Mac IIci. Small scale shared memory multiprocessor PC's are becoming more available. The 486 is itself considered a performance contender with RISC processors, and RISC co-processors, especially 860 and 88000, are being provided in personal computer systems.

All of this has been accomplished with very strict upward compatibility constraints. Binaries for the original PC, including those with direct device access, are still expected to run on a 486 multiprocessor machine. Devices built for the original PC I/O bus are still expected to plug into current high-end machines. There is only one notable exception that has been successful: IBM consciously broke compatibility with the PC/AT bus when it introduced the Micro-Channel. That introduction has led to controversy in the industry that still remains to be solved. Strict upward compatibility has also been characteristic of the Macintosh family.

Evolution Of PC Operating Systems

As the machines based on these designs have rapidly grown in performance and capacity, the limitations of the original operating systems have become fundamental barriers to progress. In the DOS environment for the PC, memory addressing assumptions are constrained by the original hardware address space, the first 640KB for DOS and applications, the next 256KB used for video buffers and other device memory, and the remaining 128KB for firmware (BIOS). Disk addressing is similarly constrained by the original small disk sizes - the standard DOS file system has 16-bit fields for identifying disk sectors, limiting the maximum file system size to 32MB. The lack of memory management in the 8088 has constrained DOS to essentially no multitasking support (except the **Terminate and Stay Resident** mechanism). Perhaps most significant is the minimal level of system services provided, because application writers have developed their own sets of system services and hardware dependencies. In the Mac environment, the addressing constraints have not been so severe, primarily due to the larger address space of the 68000, relative to the 8088. In general, there has been a richer set of systems services, leading to consistency amongst applications, and the initial inability to add hardware devices has given application writers less motivation for device-dependent code. But, the lack of memory management in the base machines prevented the development of general multitasking support.

There have been ingenious workarounds for many of these problems. In the DOS environment, the Lotus-Intel-Microsoft Expanded Memory Specification has provided for larger physical memory addressed by overlaying address ranges in the region above 640KB. "DOS extenders" for 386 machines allow applications to switch to 386 virtual mode, exploiting the power of that processor, yet switching back to DOS in a controlled manner for file services, etc. Environments such as Microsoft Windows allow for multiplexing of applications. In the Mac environment, the MultiFinder allows similar capabilities. These are still workarounds for difficult problems, so constraints such as the 640KB memory assumptions in the PC and interference between applications under the MultiFinder (T. Hogan) are still evident. However, commercial rewards of further improvement are great, and additional breakthroughs are likely to occur.

Given all of these problems, and given the tremendous potential of the hardware, it is tempting to abandon the original operating system assumptions and start over. But, it is not commercially feasible to simply start over; some level of compatibility with existing object code and conventions is a fundamental requirement due to the tremendous base of existing application software. In the PC environment, OS/2 (originally known as DOS 5.0 by its developers) preserves a high degree of DOS compatibility while attempting to exploit the hardware potential (G. Letwin). In addition, OS/2 provides relatively advanced operating system services, e.g., dynamic binding and an execution model based on threads. Dramatic enhancements in system services are anticipated in System 7.0 for the Macintosh.

There is great potential for technical success in these efforts if an acceptable balance between compatibility and new function can be found. Substantial development resources are being applied, and there is the opportunity for consistency and compactness in that the source code is essentially under the control of a single organization. OS/2 has been less influential than many anticipated. Future success of OS/2 and Mac System 7.0 will depend to a great degree on how well the developers can incorporate attractive new functions while preserving sufficient compatibility with existing software. This is especially critical with OS/2, which does not yet exploit the 386 or 486 (so that compatibility with the 286 can be preserved).

Growth Of The UNIX Operating System

Variants of the UNIX operating system have become the de-facto standard on workstations and other machines. Though implementations of the UNIX operating system have been available on personal computers for years, UNIX has not yet been a major factor in the personal computer industry. This is changing! UNIX is becoming a major force in PC's for a number of reasons, including:

• The availability of machines with reasonable memory management, e.g., the 286, and especially the 386, 486 and 68030, makes implementation of UNIX practical.

• The limitations of other operating systems, as already described, have led users to seek higher function software.

• Software developers are providing bridges between UNIX and other operating systems, so investment in applications can be preserved.

• UNIX has matured in terms of networking and user interface support.

• The conflict and consolidation in the UNIX industry associated with the formation of the Open Software Foundation and UNIX International has attracted the attention of the personal computer marketplace.

Major personal computer manufacturers, including Apple, Compaq, Dell, Everex, Hewlett-Packard, IBM, Intel and Olivetti are actively developing UNIX environments for their machines.

However, there are still serious obstacles to surmount before UNIX becomes, achieves its potential in the personal computer environment. Most of the issues for UNIX on larger machines are issues for personal computers as well, but emphasis and significance vary.

Standardization And Compatibility

One of the fundamental requirements in the personal computer environment is compatibility. The variations between versions of UNIX are grudgingly accepted in larger machine environments and are simply unacceptable in the personal computer environment. Standards such as 1003.1, NIST 151 and XPG-3 are necessary but not sufficient. Fairly strict binary compatibility is mandatory in the personal computer environment, so it will exist in one form or another. (The binary level compatibility that exists in the personal computer environment is often cited as motivation for ABI's and ANDF's). The major UNIX products for the 386 are based on system V.3. Now that V.4 for the 386 is complete, it is reasonable to expect that it will become well entrenched in the 386 marketplace. Any other commercial implementations will have to provide strong compatibility with V.4.

Application Availability

A larger machine may be commercially viable, at least at introduction, with relatively few optional applications. The traditional UNIX utilities and software development environment are certainly enough of a bootstrap for application porting and development, and the customer may have enough unique source code that this environment is sufficient. Though personal machines may be used in exactly this same fashion, the commercial volume of such usage is insignificant relative to more common applications such as document preparation, spread sheet generation, data management and computer aided design. It has been anticipated that applications from more traditional personal computer environments would become readily available in the UNIX environment, but, like the adoption of OS/2, this is happening more slowly than many anticipated. On the other hand, applications originally developed for larger UNIX machines are rapidly becoming available on personal computers as the power and capacity of these machines becomes more apparent.

User Interface

Ignoring the controversies of intellectual property and attribution, it is clear that the Macintosh has set a commercial standard of ease of use that is aspired to for other personal computers, workstations and larger machines. Microsoft Windows and Presentation Manager are addressing this in the DOS and OS/2 environment, respectively. Having abandoned *command.com* and *dir*, the PC user is not likely to accept the shell *ls-l* of UNIX. With the X Window System bandwagon, the UNIX community seems on the verge of taking the next steps (pun intended) toward the commonly accepted toolkits and desktops necessary to escape the traditional command-oriented user interface. But, this is happening far more slowly than predicted, and true X-based applications are far rarer than many hoped.

System Administration

The difficulty of system administration, beginning with initial installation and configuration and continuing with routine maintenance and trouble shooting may be the Achilles' heel of UNIX. This is especially evident in the personal computer environment, since the machine user nearly always will be the one responsible for system administration. It is critical that system administration be simplified/eliminated in so far as possible. A number of mechanisms can be applied, such as factory installation and configuration tailored to the customer and provision of remote and on site support. Better technology and consistency in system administration will continue to be a critical need.

Other issues for UNIX on personal computers receive relatively lesser attention on larger machines.

Compatibility With Other Operating Systems

One of the big advantages of UNIX on the 386 or 486, relative to other processors, is the ability to provide a very rich DOS compatibility environment which allows execution of standard DOS binaries. This is dependent on the virtual 8086 support provided in the 386 architecture, and on software such as Merge or VP/ix. Programs such as **Flight Simulator**, which are considered tests of compatibility in new hardware implementations, can execute in these environments. Data interchange is straightforward, and more complex interactions, e.g., mixing of DOS and UNIX commands in a pipeline, are practical. Unlike OS/2, which allows only one DOS compatibility instance because of 286 limitations, multiple DOS sessions are both possible and useful in a 386 UNIX environment. The level of compatibility and coexistence is markedly higher than analogous situations in larger machines, e.g., with regard to VMS on a VAX or CMS and MVS on a 370.

Full Function "Standard" Applications

The tradition and continuing usage of character terminals for UNIX has led to standard applications providing relatively primitive levels of function that would be considered substandard in the personal computer environment. For example, most UNIX mail systems deal only with ASCII text. In a typical DOS mail system, built-in support for transferring binary files, displaying images, using color to highlight text, etc. are standard.

Redundancy/Eccentricity

For many, one of the virtues of UNIX is the plethora of tools available for any given task. But for the new user, it may not be at all obvious whether *cat, more* or *pr* is the best way to display a file, nor why one would choose to use */bin/mail* over */usr/ucb/ mail*. For the DOS user, it is not obvious why the analog of the DOS *find* command is named *grep*.

Resource Consumption

One of the obstacles to acceptance of OS/2 is relatively high minimum memory and disk requirements, compared to DOS. Memory and disk requirements for UNIX are even higher. This can be dealt with, somewhat, by partitioning commands and libraries into installable subsets, but there is no generally agreed way to partitioning, and even the attempt to partition brings out a backlash from those who want the entire system.

General Trends – Micro-Kernel Implementations

Growth in UNIX is also constrained by compatibility with previous versions. As new features are added, old features are retained for compatibility, and the sum loses the compactness of the early versions. To a certain extent this can be dealt with above the traditional kernel by providing multiple library interfaces to a multipurpose system call. But even so, current UNIX kernels are often on the order of a megabyte for code and data, ignoring buffers. This is enormous compared with early kernels and substantial relative to typical personal computer physical memories. In order to reverse this trend, a number of research efforts are implementing "micro-kernels" with the intention that major system functions such as file system and network support be provided outside the kernel. This notion is not new, indeed it was part of the origin of UNIX. What is new are the objectives being pursued relative to UNIX and the other goals of these projects.

Mach

The Mach (M. Accenta, R. Baron, W. Bolosky, D. Golub, R. Rashid, A. Tevanian and M. Young) project is a descendent of the previous ACCENT project at Carnegie-Mellon University. Like ACCENT, it is designed to support shared memory multiprocessors using copy-on-write techniques and message passing. However, where ACCENT was only "UNIX-like," Mach is designed to allow strict UNIX compatibility. The versions of Mach currently available, either in products such as the NeXT systems or in research vehicles, incorporate UNIX compatibility, including the file system, as part of the kernel. This is the version now planned as the basis for OSF/1. As part of continuing Mach development at CMU, there is a new version 3 which leaves most system services outside of a small kernel providing virtual memory and message passing services as a basis for implementing other system services *above* the kernel.

CHORUS

CHORUS (M. Rozier, V. Abrossimov, F. Armand, I. Boule, M. Gien, M. Gillemont, F. Herrmann, C. Kaiser, S. Langlois, P. Leonard and W. Neuheuser) began as a research project at INRIA and led to a commercial venture, CHORUS systems. It is now in its fourth version, which is a micro-kernel version, where the kernel primarily provides virtual memory, real-time services and message passing services. At first glance, there are many analogues to the Mach project, but there is more emphasis on distributed system issues and real-time characteristics. Like Mach, there is provision for strict UNIX compatibility. OSF has indicated their intent to evaluate CHORUS, Mach 3 and other micro-kernel architectures as the basis for future operating systems.

AIX Version 3

Another approach to the size of current traditionally structured kernels is to maintain essentially the traditional content but to implement the kernel so that the majority of the kernel itself can be paged. This has advantages of preserving a more familiar structure and disadvantages relative to the difficulty of developing code for a traditional kernel environment, e.g., the typical lack of protection of one kernel portion to another. AIX Version 3 takes this approach, along with facilities for dynamic binding of many traditional services and new subsystems. This provides essentially the same flexibility advantages as a micro-kernel.

File Systems

In addition to revisiting the traditional kernel implementations, it is time to revisit file system structure and implementation. Database logging technology can be used to improve recoverability and increase performance by avoiding unnecessary writes, as illustrated in the AIX 3 file system. Support for file systems spanning volumes and mirroring will soon be considered mandatory. Many other extensions, e.g., mapped file support and contiguous file services for real-time applications, will become commercial requirements as well.

Multiprocessing

Traditional kernel implementation, particularly the lack of granularity of locking, is at odds with general multiprocessor support. Many vendors have provided multiprocessor kernels with finer granularity locking retrofits. However, none of these kernels has achieved wide multi-vendor support. There does seem to be some convergence likely now. Corollary, Compaq, SCO and others are advocating a multiprocessor variant of SCO UNIX. Though this implementation is based on V.3, there is no obvious reason why it could not be applied to V.4. It is widely believed that need for multiprocessor support was fundamental in OSF's switch from AIX 3 to Mach as the base kernel technology for OSF/1. Certainly, USO and UNIX International will have to provide some multiprocessor implementation of V.4.

Distributed Systems

For all benefits of a personal machine, most serious usage depends on sharing and cooperation amongst multiple users, and/or access to shared resources on larger machines. To retain the benefits of personal machines, distributed system technology is required to bring the entire collection of machines together. The distributed system issues have all of the compatibility problems previously cited, all of the requirements of generality previously cited, plus major new requirements.

Heterogeneity

In many environments, the investment in existing machines and the requirements for the unique capabilities of distinct types of machines mandate major differences amongst machines in their hardware and operating systems.

The current market share leader in the PC environment, Novell Netware, provides not only support for DOS and OS/2 systems, but also Macintosh systems as well. Novell is advocating a new implementation, Portable Netware, which is designed to work in UNIX, VMS and other environments. Microsoft's Lan Manager, and the LM/X product for UNIX, have similar ambitions.

The only current UNIX-oriented system with comparable goals of heterogeneous system support is Sun Microsystems' NFS. Not only is NFS the de-facto standard for UNIX, but there are NFS implementations for DOS, for the Macintosh, for VMS, for CMS and MVS. Any new distributed system implementation will have to strive for heterogeneous system support comparable to Netware, Lan Manager and NFS if it is going to have a chance of displacing these.

Transparency

For a distributed system to be effective, it must feel like one system to the end users, not an ad hoc collection of machines. This is difficult even when considering relatively homogeneous machines, simply because of performance, semantic, availability and administrative boundaries. In trying to deal with heterogeneous systems, the differences between systems make transparency much harder to achieve. Netware provides a relatively transparent environment amongst DOS machines, but is less effective in bridging to Macintosh and UNIX environments. Similarly, NFS deals relatively well with UNIX environments and less well with others.

Conclusion

There is a tremendous tension between these requirements. Progress in distributed systems requires not only advances in base technologies, but a reasonable equilibrium in resolving these tensions. Though there are numerous efforts ongoing, major progress in this area will be exceedingly difficult to achieve. The design problem is essentially that of providing a new operating system that not only solves fundamental distributed system problems, but provides compatibility with several variants of UNIX, DOS, OS/2, Macintosh and other systems. Since bringing UNIX variants together continues to be a difficult struggle, at best, it seems unlikely that distributed system technology improving on function, heterogeneity and transparency will appear soon. Yet, this is exactly what we need! ✦