Copyright © 1994, by the author(s). All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

by

Brad Peters, William R. Bush, and A. Richard Newton

Memorandum No. UCB/ERL M94/3

31 January 1994

by

Brad Peters, William R. Bush, and A. Richard Newton

Memorandum No. UCB/ERL M94/3

31 January 1994

by

Brad Peters, William R. Bush, and A. Richard Newton

Memorandum No. UCB/ERL M94/3

31 January 1994

ELECTRONICS RESEARCH LABORATORY

College of Engineering University of California, Berkeley 94720

by

Brad Peters, William R. Bush, and A. Richard Newton

Memorandum No. UCB/ERL M94/3

31 January 1994

ELECTRONICS RESEARCH LABORATORY

College of Engineering University of California, Berkeley 94720

Microsoft Windows NT And The Competition for Desktop Computing

January 1994

Department of Electrical Engineering and Computer Sciences University of California Berkeley, California 94720

Abstract

This report contains two papers, An Introduction to Microsoft Windows NT And Its Competitors, and The Status of Windows NT and Its Competitors At The End of 1993.

The first paper, written in April 1993, presents an overview of the technology of Windows NT, and analyzes the competitors and competitive factors in the desktop operating system race. It compares systems from historical, commercial, and technical perspectives.

The second paper, written in December 1993, describes the status of the competition as of that date. It tracks the early market response to NT, the products available and planned for NT, and the status of NT's competitors.



An Introduction to Microsoft Windows NT And Its Competitors

Brad Peters William R. Bush A. Richard Newton

April 1993

Department of Electrical Engineering and Computer Sciences University of California Berkeley, California 94720

As computer hardware becomes ever more powerful and less expensive, it is causing two key types of computing platforms with origins in the 1970s to converge: the large, compute-intensive, multi-purpose, multi-tasking, multi-user system and the small, inexpensive, single-user system oriented toward small interactive applications. This convergence will ultimately pit UNIX, IBM's OS/2, Apple's Mac OS, and Microsoft's Windows NT in a battle for operating system preeminence. The outcome of that battle will determine the dominant user interface and system paradigms for the next decade.

This document performs four functions.

- It gives a brief history of the competing systems, including current products, in Section 1.
- It presents the technology of Windows NT. Specifically, in Sections 2 and 3 it outlines the technical characteristics of NT as described by Microsoft. Note that it does not compare these characteristics with those of UNIX or other operating systems, nor does it evaluate the actual NT implementation, available now only in beta test.
- It describes other relevant but perhaps unfamiliar technology: OS/2 in Section 4 and future object oriented operating systems in Section 5.
- It compares NT with UNIX and other operating systems, in the broader commercial context, in Section 6.



1. Background and History

This section reviews the evolution of the emerging primary operating system contenders, and highlights current products.

1.1. The Evolution of UNIX

. ---

1

In 1969 Bell Laboratories withdrew from a joint project with General Electric involving a multiuser interactive operating system called Multics. About the same time Ken Thompson of Bell Labs began tinkering with a Digital Equipment Corporation PDP-7 minicomputer with the official aim of creating an operating system that could support the coordinated efforts of programmers in a research environment. With the strong influence of Multics the UNIX operating system was born (in fact, UNIX is a play on the word Multics). In order to satisfy management, Thompson proposed that further development of UNIX be supported by Bell Labs in order to provide a document preparation tool for the Bell Laboratories patent organization. In fact, an early UNIX system using a PDP-11/20 was actually delivered to the Bell Laboratories patent office in 1971. This original marriage of the operating system with the needs of sophisticated (type-set) document preparation served to make UNIX rather general-purpose.

Ken Thompson's original development efforts resulted in an operating system, an assembler, and several assembly language utility programs. However, in 1973 Dennis Ritchie took an interest in the operating system and transformed it dramatically by rewriting the operating system in his new general purpose language C, which he designed specifically to work on the UNIX. C evolved directly from Ken Thompson's B, which in turn evolved from Martin Richard's BCPL, a systems programming language. C has proved to be very adaptable to many different types of computer architectures.

With over 90% of the operating system kernel and the vast majority of utility programs written in C, the UNIX operating system is highly transportable. This disconnecting of the operating system from its development machine allowed Ritchie and Stephen Johnson to complete the first port of UNIX in 1976 to the Interdata 8/32. Since then, UNIX has been ported to virtually every popular computer architecture ranging from single chip microprocessors to large mainframe computers.

In July of 1974 Ken Thompson and Dennis Ritchie published their classic paper "The UNIX Time-Sharing System" [3], and launched wide-spread interest in UNIX, especially when people discovered that for just \$150 they could acquire an unsupported copy of Version 5 with complete source code. By 1976 Version 6 was distributed with a shell and over 100 utilities very similar to most modern day versions. In 1978 Version 7 was released by Bell Laboratories, featuring the Bourne shell and other enhancements.

During the late 1970s and early 1980s, while AT&T had virtually stopped developing UNIX, it was aggressively supported and improved by a group of graduate students at the University of California at Berkeley. Starting with the AT&T release, Berkeley students created BSD 3 and BSD 4 (BSD stands for Berkeley Software Distribution). With the addition of the C shell, the visual editor vi, the Franz Lisp version of LISP, Berkeley Pascal, networking support, improved interprocess communication via sockets and pseudo-ttys, virtual memory support, and many significant performance enhancements, BSD UNIX systems became extremely popular and helped vault UNIX into the realm of truly modern operating systems.

Recognizing the value of the Berkeley effort, AT&T incorporated many pieces of the Berkeley work into its own product. Other vendors, most notably Sun Microsystems and Digital Equipment, opted to use Berkeley's version directly. Although BSD included AT&T code, that code was from an older, less expensive version of UNIX. In fact, as AT&T raised license fees and imposed increasingly restrictive terms, it forced commercial implementers to look for alternatives. The Open Software Foundation was born, with key OSF members including Hewlett Packard, DEC, IBM, and Apollo, intending to develop a UNIX-like operating system that did not require paying AT&T license fees (and that would keep AT&T and then-collaborator Sun from establishing a UNIX monopoly). Out of OSF came Motif, a commercial X-window tool-kit, and OSF/1, a mix of Carnegie-Mellon University's Mach kernel and IBM's AIX version of UNIX. OSF served its purpose; UNIX Systems Laboratories, the former AT&T subsidiary responsible for UNIX, completely revamped its licensing policies.

There is an inherent tension between UNIX's openness for users -- standard functionality -- and the need for commercial vendors to differentiate their products. There is a similar tension between UNIX's

openness for users and its technological openness (via readily available source code) to improvement by academic and commercial tinkering [26].

Value-added specialization and differentiation have come to dominate. Users no longer get UNIX from AT&T or Berkeley. Instead, third party vendors market their own versions. Sun, DEC, HP, IBM, Silicon Graphics, and CRAY, for example, each provide versions of UNIX for their hardware platforms. Various versions exist for Intel-based hardware, such as Xenix, Santa Cruz Operation, Interactive Systems, Microport, and BSDI. In 1989, the DMR Group found about 25 brands of UNIX as the result of a survey of about 6000 sites [30].

USL, now part of Novell, sells the "original" AT&T System V. The new version, System V Release 4.2, also known as Destiny, breaks out Berkeley compatibility as a separate offering. As part of Novell, USL has a solid desktop distribution channel. NetWare, which currently holds the vast majority of the personal computer networking market, is now bundled with SVR4.2 to create a product called UNIXWare. This allows users to access UNIXWare file and print servers without going through a TCP/IP gateway.

Santa Cruz Operation (SCO) has stayed with SVR3.2. Open Desktop 2.0, SCO's flagship UNIX product, is aimed at developers who want a stable, proven UNIX. According to Unigram-X newsletter editor Maureen O'Gara, though, "The conventional wisdom says that Sun, Univel, and so on are going to eat [SCO's] lunch" [30].

Sun is promoting Solaris 2. Solaris is a derivative of SVR4, with a few special features such as symmetric multiprocessing. Sun sees SMP as a key new technology in the next few years, and has developed hardware (such as the SPARCstation 10) and software that can take advantage of it. However, Sun has alienated many of its users by abandoning BSD. The change to an SVR4 derivative was probably inevitable, but Sun made the transition more painful than necessary, especially by unbundling its compilers and selling them through a separate vendor. SunSoft has also so far not agreed to follow OSF's Distributed Management Environment (DME), an operating system-independent network model for system administration; this issue is being addressed by the newly formed COSE effort, described below.

Despite this specialized commercialization, unlike DEC's VMS or IBM's OS/2, UNIX is a powerful multi-platform operating system that is relatively free from any one hardware vendor. In addition, its origins as a product of research have given it a rich set of features that have evolved as a response to the needs of programmers and sophisticated users. UNIX's openness has led to such wide distribution that a large body of software and experience is based around it.

UNIX standards are being developed. Standards organizations include X/Open, UNIX International, and the IEEE. The IEEE's POSIX standard [31] defines basic operating system functionality, enhanced functionality for certain processing domains, and programming language interfaces. POSIX is, roughly, an amalgam of the core parts of USL's System V Release 3.X and BSD 4.3. In addition, System V Release 4 is becoming a de facto standard. DEC and Sun still maintain BSD versions of UNIX (Ultrix and SunOS Version 4), but are moving to SVR4 with OSF/1 and Solaris 2. And unlike SVR3.2, SVR4 makes networking and graphics standard components. MIT's X Windows is now standard with OSF's Motif and SunSoft's OPEN LOOK (and other versions of UNIX) as companion window managers and GUI toolkits.

In addition, UNIX vendors are coming to understand that the challenge from Microsoft is serious, and that a major component of that challenge is a truly uniform portable operating system. In response HP, IBM, SCO, SunSoft, Univel, and USL recently announced a Common Open Software Environment (COSE).

1.2. The Evolution of the Macintosh

In the 1970's a group of researchers at Xerox's Palo Alto Research Center (PARC) developed the first user friendly workstation technology. They produced the Alto, a relatively powerful small computer (for its day), with a mouse, bitmapped display, and software that took advantage of the hardware platform (including an editor similar to Microsoft Word). Xerox was unable or unwilling to commercialize the technology. Apple, looking for new products different from the Apple II, stepped in and began developing products based on PARC's technology.

The first product was the Lisa; development started in 1979, and the product was introduced in 1983. It was not successful, being too expensive with too little software. Its relatively sophisticated graphical user interface was not enough to sell it.

The second product was the Macintosh, released in 1984. It was designed to be cheaper and easier to use, a computer appliance for non-experts. Partly through lessons learned in engineering the Lisa, the Mac was successful, providing a more expensive but more sophisticated alternative to the IBM PC.

From the beginning the Mac and the Lisa were conceived as a combination of proprietary hardware and software. Unlike the PC, there was to be no clone market. Indeed, bitmapped displays and mice were uncommon in 1984, and such hardware was special. Apple could charge a premium for it, and could induce software developers to write software for it.

As time went on, however, and as hardware became cheaper and as others realized the value of a graphical user interface, the overall Mac technology became less revolutionary. Macintosh development has become focused on price/performance improvements, keeping its proprietary technology competitive with developing PC and workstation technology. Apple has to a large extent lost its monopoly over the PARC paradigms. There is still a large base of excellent application software written for the Mac platform, but its proprietary hardware and nonportable system software, combined with the difficulty of writing applications for it, have restricted its popularity.

Not long ago Apple released a new version of its system software for the Mac, System 7. This software adopts features from modern operating systems, and keeps the Mac technology current. But it does not give Apple a competitive advantage. The only new technology on the Apple horizon is the new operating system being developed by the Apple-IBM joint venture Taligent, discussed below. This technology is completely unproven.

1.3. The Evolution of Windows NT and OS/2

Whereas UNIX has its roots in the mainframe and minicomputer industry, Microsoft Windows NT has evolved from the opposite end of computing, the low-end personal computer. By the mid 1970s microchip manufacturers had begun mass producing single chip 8 bit microprocessors. From Motorola's 6502 to Intel's 8080 and Zilog's Z80, there was enough power in a single chip to allow the construction of personal computers.

The initial personal computers were slow and not very powerful. Most used the CP/M operating system; some simply used a library of utility routines for managing floppy disk files. CP/M provided a basic suite of functions that enabled programs to do simple console input/output and file management. The machines were single user, and usually programs made direct use of all hardware facilities. Most programs were written in assembly language to optimize for speed and code size (typically the microprocessors could address only 48K-64K of RAM).

In the late 1970s, Intel produced the 8088 and 8086 16-bit microprocessors. They represented a significant leap in processor power and allowed much larger and more sophisticated programs to be written. Accordingly, IBM decided to enter the personal computer market in 1978. The result was its highly successful IBM Personal Computer. With its open architecture, third party vendors were free to create add-on boards and hardware to enhance the IBM PC's functionality. Further, vendors were free to clone the IBM PC and offer their own personal computers. The initial machines, which sold for several thousand dollars and came with 128K RAM, two floppy drives, and an 8088 CPU running at 4.77 MHz, represented a substantial leap in computing power.

IBM had originally planned to offer a 16-bit version of CP/M with its new personal computer. However, a combination of clever marketing by Bill Gates and Microsoft Corporation, and technical blunders by Digital Research, the maker of CP/M, convinced IBM executives to use an operating system from Microsoft (see [12]). The first version of MS-DOS, version 1.0, was basically a quick rewrite of CP/M, substituting system interrupts for system calls.

Gradually, with the release of later versions of MS-DOS, users were provided with support for hard disks, hierarchical file directories, block and character device drivers, and a suite of file handling utilities.

However, MS-DOS always remained more of a library of utility routines than an active operating system. Despite the increased power of the 8086 over its predecessors, programs still made direct use of hardware to optimize efficiency. Moreover, inadequacies of the single user, single task operating system that relinquished all control over the machine when programs were executing led to such programming anomalies as Terminate and Stay Resident programs (TSR's) that attached themselves to the end of the operating system and waited to be activated, and interrupt hooks that would intercept all incoming hardware interrupts before the system would receive them. The speed advantages of allowing the programs to have complete control over the system led to sloppy programming with a wide variety of incompatibilities among various PC platforms.

Intel later introduced the 80286, a full 16-bit multi-tasking microprocessor that provided all of the necessary architectural features required by a modern operating system. Indeed, with the new chip executing powerful instructions in fewer cycles at 10 to 16 MHz, Microsoft was able to port to the PC a version of UNIX called Xenix. However, the computers were still not powerful enough to be truly multi-user, and the overhead of a large operating system sapped the power of the machines. Moreover, the IBM PC and PC compatible market had ballooned and there was now a large amount of MS-DOS-based software that people wanted to use.

With the introduction of the 80386 by Intel, a new door was opened. With the advent of the virtual 8086 execution mode, users could run several MS-DOS tasks simultaneously. Also, the 80386 was a full 32 bit microprocessor, faster than the 80286, and it could run at clock speeds of up to 33 MHz, making it 30 times faster than on the original IBM PC.

When the Macintosh was released in 1984, it provoked concern at Microsoft, and Gates realized that graphical user interfaces would be the future of computing. Accordingly he started development of a graphical user interface that would run on top of MS-DOS and provide an abstraction layer between it and application programs. Microsoft Windows version 1.0 was hardly a success. Its primitive user interface and lack of support for MS-DOS programs left users unwilling to abandon their old software. Its device independent input/output routines were useful for a few applications, such as Aldus Pagemaker, a desktop publishing package, but most of those applications were shipped bundled with run-time versions of Windows.

With Windows 2.0 Microsoft introduced several Windows varieties. There was a version for 8086based machines that was roughly similar to version 1.0 with a few enhanced user interface features. At that point, though, the 8088 was becoming a relic and most of the personal computers sold contained the 80286 or 80386 microprocessor. Accordingly, Microsoft also introduced Windows 286 and Windows 386, each taking advantage of the enhanced capabilities of the host microprocessor. With Windows 286, users had a faster CPU and could access much larger pools of memory. However, with Windows 386, users could run DOS shells in a window. That proved to be a major advantage.

In 1989, Microsoft released Windows version 3.0. Finally users had an advanced multi-tasking graphical user interface for their personal computers along with the hardware to make it work -- when it was run on a 80386-based computer. Users could run a large set of Windows-based software, as well as all of their MS-DOS software in special windows emulating virtual PCs. It finally provided the first credible graphical user interface for the IBM family of personal computers. After a year Microsoft was selling up to many thousands of copies a day. Thousands of programs were written for Windows and an entire software industry sprang up overnight.

In April of 1992, Microsoft released its next upgrade to Windows, version 3.1. This included many enhanced features such as vector scalable fonts, a more sophisticated error handling mechanism, an improved shell, multimedia extensions, pen computing extensions, as well as several object-oriented features such as an Object Linking and Embedding (OLE) protocol (which allows files to contain pointers to other files) and a drag and drop mechanism. In addition, Windows 3.1 was still able to run software written for version 3.0. By August of 1992, Windows version 3.1 was selling at a rate of 1 million to 1.5 million copies a month [20]. Moreover, InfoCorp Computer Intelligence estimated that by August there were 10 million Windows users and expected that 11.1 million copies will be in use by the end of 1992 [30]. This is about a tenth of the estimated 100 million personal computers in use [30].

The success of Microsoft Windows, and indeed the strategy behind Windows NT, are based somewhat on lessons learned from OS/2. When the Intel 80286 was released in 1983 only MS-DOS was available, which was unable to take advantage of the sophisticated features of the new processor. Accordingly, Microsoft and IBM teamed up to develop the next generation personal computer operating system, OS/2. IBM released the initial version, 1.0, in 1987, to a lukewarm reception. It had many flaws that kept it from gaining wide acceptance.

Version 1.0 of OS/2 was a truly multi-tasking operating system. Unfortunately, though, the presentation manager user interface was new and there were very few programs available for it. Since the operating system was tied to the 16-bit Intel 80286 architecture it could not take advantage of the newer Intel 80386's virtual 8086 mode -- OS/2 1.0 allowed the user to execute only one MS-DOS program at a time.

Version 1.0 also imposed minimum hardware requirements that at the time were normally satisfied only by higher end workstations. Several megabytes of RAM were needed for the system to operate efficiently when RAM cost \$400 per megabyte. Moreover, the multi-tasking power of the operating system was not matched by the 80286 CPU performance. When several applications ran simultaneously, performance was weak. Although the system was sound, it never caught on. OS/2 1.0 was used mostly as a platform for multi-tasking network file server software, such as Microsoft Lan manager, Novell NetWare, and Banyan Vines.

After its initial release, IBM and Microsoft began working on the next version of OS/2. The new version was to be a full 32 bit operating system that would operate exclusively on the 80386/486 architecture. During the development of OS/2 2.0, IBM and Microsoft also started planning the features and design of OS/2 3.0, a truly next generation transportable 32 bit operating system. However, during this discussion process, IBM and Microsoft clashed on design philosophy. Most importantly, Microsoft wanted OS/2 3.0 to use an enhanced Windows Application Programming Interface. After a bit of turmoil IBM and Microsoft decided to go their separate ways, and Microsoft promptly renamed its OS/2 3.0 development project Windows NT, or "New Technology." IBM decided to enhance OS/2 2.0 and release it ahead of Microsoft in a bid to capture the next generation personal computer operating system market.

Since the IBM-Microsoft rift IBM has produced an operating system with more than 2 million lines of code, developed by 600 programmers. Estimates put IBM's investment in OS/2 at well over \$1 billion[21]. Despite this substantial commitment, some still question IBM's resolve, citing its lackluster marketing and serious involvement in Taligent, the joint IBM/Apple venture aimed at producing a 32-bit objectoriented operating system called Pink [15]. However, analysts now suspect that IBM is beginning to use Taligent more as a research institution than a viable development company. Recent reports suggest that IBM is taking much of what it is learning from Taligent and attempting to incorporate that into future releases of OS/2 [30]. In contrast, IBM has recently been much more aggressive selling OS/2 2.0, offering it as essentially an upgrade from Windows or MS-DOS. IBM reduced the suggested retail price from \$500 to \$195, and then began offering a direct price of \$79 to current Windows users, \$99 for MS-DOS users, and \$149 to new users. It has decided to market OS/2 at prices that clearly do not cover its development costs, in a strategic bid to capture as much of the market as possible before Microsoft is able to finally release Windows NT.

Currently, IBM claims that over 1 million copies of OS/2 have been sold. This compares, however, to the 11 million copies of Windows that have been sold in the last two years [20].

To avoid the all-or-nothing hardware specific boxes that IBM has built for itself, Microsoft has created a hierarchy of Windows products designed to fit the needs of users in the broad spectrum from small, low-performance personal computers to high-end workstations and multiprocessing servers, including non-Intel-based hardware.

The first product, basic Microsoft Windows 3.1, is aimed at stand-alone users. It operates on top of the MS-DOS operating system.

The second product is an extension of Windows 3.1 and is designed to support the small workgroup. Windows for Workgroups adds networking features to the Windows 3.1 platform. Users can send each other electronic mail and share each other's file systems and printer resources. There is a limited security system that requires users to have passwords in order to gain access to portions of file systems or particular print-servers. Moreover, users can control the extent to which Windows allocates CPU usage to local tasks versus remote requests. Perhaps the most powerful component of Windows for Workgroups is networked Object Linking and Embedding (along with Dynamic Data Exchange, both discussed below). Applications that reside on one machine can be accessed as servers, managing data embedded in documents created by applications on other machines.

The next level in Microsoft's product hierarchy is Microsoft Windows NT itself. Indeed, it is not only a more expensive product, but it demands considerably more from the host hardware than does Windows 3.1. NT requires more than double the memory and will not operate at a usable speed without at least a 30Mhz 80386 or faster CPU. Networking is built in with both peer-to-peer services as in Windows for Workgroups and with the ability to use NT servers in single-domain networks. It also includes features for distributed computing such as remote procedure call and support for TCP/IP.

The final stage in the Windows product family is aimed at administrators of large multiserver networks. The Advanced Server Version of Windows NT is expected to include tools for network management such as RAID 5, drive duplexing, and disk mirroring.

In the longer term (approximately two years), three Windows products are likely: Modular Windows, a reduced version for consumer electronics and real-time applications; Windows 4.0, which will contain much of the functionality of Windows for Workgroups; and Windows NT, for users requiring the networking and security features of NT.

2. Windows NT Technology: Basic Decisions

This section and the next describe Windows NT technology in broad terms. These descriptions are not comparative or critical analyses. They are instead aimed at presenting the information Microsoft has made available about NT, to give a sense of what Microsoft hopes to accomplish with the system (consider it a presentation of plausible motherhood). A critical analysis of the technology would be an appropriate follow-on to this document after actual experience with the NT system.

Microsoft Press recently released a book detailing the features of Windows NT, *Inside Windows* NT, by Helen Custer [6]. The material in this section and the following one is condensed from that book.

This section outlines the basic goals and design decisions behind NT. The next section presents an overview of the NT implementation.

2.1. Basic Requirements

The NT team (headed by David Cutler of VAX VMS fame) identified five strategic requirements for the new system:

Portability

Learning the lesson of UNIX, the team realized that a long-lived operating system capable of taking advantage of the fastest and cheapest hardware must be portable to a wide variety of computer architectures.

Multiprocessing and Scalability

Recognizing that current hardware technology may be nearing speed limits and that concurrency may be needed to realize further increases in performance, the team planned a system that could easily execute applications on more than one processor.

Distributed Computing

Again learning the lesson of UNIX, with advances in the processing power of desktop computers, the rapid improvement of networking technology, and the massive proliferation of cheap personal computers, the team determined that a well integrated and powerful set of networking facilities built into the operating system was necessary.

POSIX Compliance

Noting the massive proliferation of UNIX in the workstation marketplace, coupled with the U.S. government procurement requirements specifying POSIX (Portable Operating System based on unIX) compliance, the team identified the value of an optional environment that could support the execution of POSIX compliant programs.

7

• Government-certifiable Security

The U.S. government specifies computer security guidelines for many government applications. The team realized that in order to compete in the government arena a certifiable security level would be necessary. The initial goal was to meet the C2 rating (with A most stringent and D least stringent), defined as providing "discretionary (need-to-know) protection and, through the inclusion of audit capabilities, for accountability of subjects and the actions they initiate."

2.2. Basic Design Goals

Before actually building the system, the team decided upon a set of design goals that would drive the development process. These goals were intended to guide the thousands of ancillary decisions that would determine the internal structure of the large project. The goals were extensibility, portability, reliability, compatibility, and performance (not necessarily in that order).

2.2.1. Extensibility

With the advent of new technology and new requirements, operating systems invariably change with time. Historically these changes have occurred incrementally with the gradual addition of new features. However, with conventional monolithic designs, programmers would simply "hack" new features into the code, causing the system to grow in an ad hoc way. The NT design attempts to avoid this paradigm.

Drawing from the Mach experience at Carnegie-Mellon University, Windows NT attempts to segregate various portions of the operating system. At the core is an operating system base, the executive, that provides primitive operating system capabilities executed in the local processor's privileged instruction mode. On top of that base are protected subsystems executed in user mode, including application programming interfaces, which provide the full set of operating system features an application requires. With this structure, the protected subsystems may be modified or replaced without affecting the stability or integrity of the protected operating system kernel.

Windows NT is also designed in a modular fashion. The executive contains a discrete set of individual components that interact with each other through a well defined functional interface. This allows new executive components to be added in a modular way without affecting the operation of existing components.

Windows NT also uses objects to represent system resources. This allows system resources to be managed uniformly through a special set of object services. Adding new objects will not undermine existing objects or require existing code to change.

The input/output system of Windows NT supports drivers that can be loaded as the operating system is running. This allows new file systems or device drivers to be added to the system by simply loading the proper driver.

A remote procedure call (RPC) facility allows an application to access remote services without regard to location on a network. New services may be added to any machine on the network and can be immediately available to applications on other machines on the network.

2.2.2. Portability

An enduring operating system should not be tied to a particular architecture. Windows NT attempts to isolate hardware dependent code to a very small and well defined segment of the operating system. It is written primarily in C, with a few portions such as the graphics component of the environment and portions of the networking user interface written in C++. Assembly language is used only for the small, well isolated parts of the system that must communicate directly with the hardware (such as the trap handler) and for components that must be optimized for speed. In addition, Windows NT encapsulates platform-dependent code (code for different versions of the same processor) inside a dynamic-link library known as the hardware abstraction layer (HAL). This layer abstracts such features as caches and I/O interrupts with a layer of software.

2.2.3. Reliability

Reliability as defined by Microsoft refers to two related ideas. First, the operating system should be

robust by responding predictably to error conditions. Second, the operating system should actively protect itself and its users from accidental or deliberate damage by user programs. Windows NT uses a method of structured exception handling for capturing error conditions and responding to them. Either the operating system or the processor issues an exception whenever an abnormal event occurs and special exception handling code is invoked.

Windows NT also benefits from its modular design. With individual components interacting through a well defined procedure call interface, components are much less likely to disrupt system integrity. Moreover, Windows NT's file system, the NT file system (NTFS), is designed to recover from all types of disk errors. It uses redundant storage and a transaction-based scheme for storing data to ensure recoverability.

2.2.4. Compatibility

Realizing that the success of an operating system depends upon the quality and quantity of applications that can be run on it, Microsoft has sought to make as many programs compatible with Windows NT as possible. Windows NT plans to be binary compatible whenever possible and source-level compatible with a wide variety of application programming interfaces. Through the use of protected subsystems, Windows NT provides execution environments for applications other than its primary programming interface, the Win32 API. When running on Intel processors, Windows NT's protected subsystems supply binary compatibility with existing Microsoft applications including MS-DOS, 16-bit Windows, OS/2, and LAN Manager. On the MIPS RISC processors, binary compatibility is achieved for MS-DOS and 16-bit Windows applications using an emulator, and source-level compatibility is achieved for OS/2 and LAN Manager programming interfaces. Windows NT also provides source-level compatibility with POSIX applications that adhere to the IEEE POSIX definition.

2.2.5. Performance

As a final goal of Windows NT development, the operating system is intended to be fast and efficient. All inter-module communication between protected subsystems and privileged code has been optimized using a high-speed message passing mechanism known as local procedure call (LPC).

2.3. Basic Operating System Models

Windows NT was designed with three operating system models in mind: the client/server model, the object model, and the symmetric multiprocessing model. Each model guided the development of the system, and each illuminates its structure and operation.

2.3.1. The Client/Server Model

The client/server model envisions the operating system as a collection of modules, each supplying services for the others and communicating through a message passing substrate. For example, memory services, process creation services, and scheduling services would each reside in a separate module. Each of these servers would run in user mode, like an application. The client, which could be either another operating system component or an application program, would request a service by sending a message to the server through a message passing kernel running in kernel mode.

This model results in an operating system whose components are small and self-contained. Moreover, because servers run in user mode, a single server can fail without crashing or corrupting the rest of the operating system. Different servers can run on different processors in a multiprocessor environment, or even on different computers, making the operating system suitable for distributed computing. The Mach operating system is a contemporary example of such a client/server approach, implementing a minimal kernel with most functions running in user mode.

An alternative to the client/server model is the layered model. In this model different operating system services are identified as being more or less basic; the basic services are put in the kernel, while the others run in user mode. Multics is the canonical example of a layered system. It in fact supported a hierarchy of kernel modes and a hierarchy of basic services.

Windows NT fundamentally uses the client/server model, but employs aspects of the layered model. It provides services such as virtual memory management and I/O handling with separate, modular client/server components. It uses a low-level kernel layer (known as the NT kernel), however, to provide low-level services, such as multiprocessor synchronization and interrupt dispatching, upon which the modular components are layered. Beneath this kernel is another, lower layer, the Hardware Abstraction Layer (HAL), which is written in assembler and is the only code that manipulates hardware directly. All of these layers (client/server modules, kernel, and HAL) run in kernel mode and constitute the NT Executive.

2.3.2. The Object Model

While Windows NT is not, strictly speaking, an object-oriented system (as defined by Bertrand Meyer in his book *Object-oriented Software Construction*), it does use objects to represent all resources that may be shared between processes. These resources include files, shared memory, and physical devices. This model limits the effects of changes to an operating system resource to the code that implements that resource's object services and its representation. Since the operating system accesses objects uniformly, creating, deleting, and referencing them through standard methods, tracking resource usage is done simply by monitoring creation and use of objects.

The object model also enhances security in the manner of earlier capability-based operating systems. All objects are protected in the same way by the security system. Every time a process attempts to access an object, the security system intervenes through the object mechanism and checks to see if that access is allowed.

The object model also provides a convenient method for allowing resources to be shared between two or more processes, since object handles (pointers) are used to refer to objects, and can be passed and shared. Furthermore, the object manager provides a uniform mechanism for managing the storage used by objects (including garbage collection).

2.3.3. The Symmetric Multiprocessing Model

Given the goal of supporting scalable multiprocessing, two approaches are possible -- asymmetric or symmetric. Asymmetric multiprocessing operating systems select one processor to run the operating system code while other processors run user jobs. Because the operating system code runs on a single processor, ASMP operating systems are relatively easy to create, simply by extending existing single processor systems. However, the ASMP approach tends to be non-portable and does not lend itself as easily to balanced resource utilization.

With symmetric multiprocessing, used by Windows NT, the operating system can be running on any free processor or on all of them at the same time. In addition to better resource utilization, SMP systems are less vulnerable to processor failure, since system code can be run on other processors if one fails. Also, since symmetric hardware is implemented relatively uniformly by different vendors, it is possible to create a portable SMP operating system.

Microsoft Windows NT also incorporates several features that enhance its performance as a symmetric multiprocessing system:

- It supports multiple threads of execution within a single process. This allows one process to execute several threads on several processors simultaneously.
- It implements server processes that use multiple threads to deal with multiple requests from more than
 one client simultaneously.
- With the exception of kernel components, it permits all operating system code to be preempted when a higher priority thread needs attention.
- It incorporates convenient methods for sharing objects between processes as well as flexible interprocess communication facilities, including shared memory and an optimized message-passing facility.

3. Windows NT Technology: System Structure

Windows NT can be broken into two parts. The first part operates in kernel mode and contains the

NT Executive. The other portion executes in user mode and includes the protected subsystems, primarily application programming interface servers.

3.1. The Windows NT Executive

The Executive is the kernel mode portion of Windows NT. It does not run in a process of its own but instead runs in the context of an existing process. When an important system event occurs, such as a system service call or an external interrupt, the Executive takes over the currently executing thread and calls the appropriate code to handle the event before returning control to the previously executing code.

The responsibilities of each of the Executive's components are outlined below. This outline is taken from pp. 29-30 of *Inside Windows NT* [6].

Object Manager

The object manager creates, manages, and deletes NT Executive objects that represent system resources.

Process Manager

The process manager creates and terminates processes and threads. It also suspends and resumes the execution of threads and stores and retrieves information about NT processes and threads.

• Local Procedure Call (LPC)

The LPC facility passes messages between a client process and a server process on the same computer.

Virtual Memory Manager

The virtual memory manager provides a large, private address space for each process. When memory usage is too high, the manager pages physical memory to disk.

• Security Reference Monitor

The security reference monitor enforces security policies on the local computer by guarding system resources and performing run-time object protection and auditing. Although this system currently complies with the government's C2 rating it may eventually be upgraded to stricter security standards.

Kernel

The kernel responds to interrupts and exceptions, schedules threads for execution, synchronizes the activities of multiple processors, and supplies a set of elemental objects and interfaces that the rest of the NT Executive uses to implement higher level objects.

• The I/O Subsystem

This subsystem is composed of a group of components that are responsible for processing input from and delivering output to a variety of devices. The components are:

• I/O manager

Implements device-independent input/output facilities and establishes a model for NT Executive I/O.

• File systems

Drivers that accept file-oriented I/O requests and translate them into I/O requests bound for a particular device.

• NT Executive device drivers

Low-level drivers that directly manipulate hardware to write output to or to retrieve input from a physical device.

Network redirector and server

File system drivers that transmit remote I/O requests to a machine on the network and receive such requests from other computers.

Cache Manager

Improves the performance of file-based I/O by storing information in a most recently used (MRU) cache. The cache manager uses the virtual memory manager's paging facility to write modifications automatically back to disk in the background.

• Hardware Abstraction Layer (HAL)

This portion of the Executive places a layer of code between the rest of the system and the actual hardware platform on which the system is running. It hides hardware-dependent details such as I/O interfaces, interrupt controllers, multiprocessor communication, and others from the rest of the system. Rather than access such hardware features directly, the Executive maintains maximum portability by calling the HAL routines when they need platform-dependent information.

3.1.1. Objects

Most of the Windows NT Executive native services are object services, performing some sort of operation on a Windows NT Executive object. Applications obtain and pass around handles to objects that represent system resources.

Shareable resources such as processes, threads, files, and shared memory are all internally represented by generic objects with a generic set of methods for manipulating them. This generality allows Windows NT to take advantage of the similarities between resources and to make the most of common code. The NT Executive object management system supports several forms of object management tasks including object naming, monitoring object limits (effectively quotas on the amount of resources available to each process), sharing objects between processes, and resource security.

For example, when a file is opened, the input/output system calls the object manager to create an open file object and return a handle to it. When an application is started, the process manager calls the object manager and creates process and thread objects with corresponding object handles. Additionally, when an object is created it can be given a name that allows it to be accessed by more than one process. A secondary process with sufficient access privileges can simply retrieve the object's name from the object manager and then open a handle to it.

Objects also provide a general method for securing all operating system resources. Whenever a process attempts to open a handle to an object, the security system is activated. Each object has associated with it an access control list that indicates which processes have access to it and what functions they may perform on it. Since an application must retrieve an object handle to access any system resource, all system resources are checked by the security manager.

3.1.2. Virtual Memory

Although each environment subsystem provides a different view of memory to its applications, Windows NT maintains a flexible underlying memory structure that environment subsystems can access using native services provided by the NT Executive.

The memory structure is a 32-bit flat memory space. Each process' virtual address space is 4 gigabytes (2^32) long. Two gigabytes are reserved for system use with the remaining 2 available to the process. The virtual memory manager pages on a least recently used (LRU) basis.

3.1.3. I/O and File Systems

The native I/O system provided by the NT Executive is asynchronous. A user can request an I/O operation and then do other processing while the operation is being carried out. The system automatically informs the caller when I/O is complete so that the caller can do any subsequent processing. I/O operations can also be done synchronously if desired, with the caller waiting until the requested I/O operation is complete.

Windows NT's modular design allows it to support a wide variety of file systems simultaneously. It supports the file allocation table (FAT) system used by MS-DOS, the high performance file system (HPFS) used by OS/2, and the new NT file system (NTFS). NTFS provides extended capabilities beyond both FAT and HPFS that include:

- File system recovery.
- The ability to handle storage media of up to 17 billion gigabytes in size.

- Security features including execute only files.
- Unicode file names that allow documents to be transferred from one computer to another internationally without losing file or path names.
- Support for the POSIX operation system environment, including hard links and case sensitive names, and information about when a file was last opened.
- Features that allow future extensibility including transaction-based operations to support fault tolerance, user-controlled version numbers for files, multiple data streams per file, flexible options for naming files and file attributes, and support for popular file servers such as AppleShare, Sun NFS, and Novell NetWare.

The I/O manager allows device drivers and file systems to be loaded dynamically into and out of the system. Drivers are modular and can be layered on top of one another. This allows two different file system drivers to access the same floppy disk or hard disk driver. Moreover, this allows new drivers, such as fault tolerant drivers, to be entered into the hierarchy at intermediate levels.

This modular view also allows access to files on the network through another file system driver called the network redirector. The redirector accepts requests for remote files and directs them to a network server on another machine.

3.2. Protected Subsystems and the Win32 API

Windows NT application servers are called protected subsystems because they are effectively isolated from other processes by the NT Executive's virtual memory system (and are not part of the Executive). Since they do not share memory they communicate through the local procedure call (LPC) message passing system.

There are two types of protected subsystems, environment subsystems and integral subsystems.

Each environment subsystem implements a specialized interface to the NT Executive, called an application programming interface (API). Each API is specific to a particular class of application platform. Windows NT provides POSIX, OS/2, Win32, and Virtual DOS Machine environment subsystems. The most important one is the Win32 API, which implements the Microsoft Windows 32 bit application programming interface. This subsystem controls NT's graphical user interface and all user input and application output.

Integral subsystems provide important operating system functions in a protected environment. The security subsystem, currently the only such subsystem, monitors the security policies in effect on the local computer. It keeps track of which users have which privileges, which system resources are audited for access, and whether audit alarms or audit messages should be generated. The subsystem also maintains a database of user accounts and accepts logon information.

The remainder of this subsection describes the Win32 API in detail.

3.2.1. The Graphical User Interface

The Windows NT graphical user interface provides a window manager that takes care of creating, destroying, iconifying, sizing, and painting windows and dialog boxes. It manages all user requests by sending messages to the appropriate windows. While it is not as aesthetically intricate as the ones provided by OS/2, Motif, or OPEN LOOK, it does provide a full-featured window and graphics manager. Aside from a few additional features such as Bezier curves, the NT API is the same as that provided with Microsoft Windows 3.1, and is based heavily on the Macintosh interface.

It makes considerable use of object handles and messages. Windows, dialog boxes, user controls, and even memory blocks are all objects managed by the API through 32 bit handles. When an application creates a window, it is given a handle to reference that window. All requests to manipulate a window, including re-sizing, managing scroll bars, and painting, must be accompanied by the window handle. Each time a user control such as a button, edit field, or scroll bar is created, it too is given a handle that must be used for future reference.

Accompanying each window and dialog box must also be a piece of code that responds to messages generated within the window or dialog. When a user moves the mouse or clicks on a button, Windows sends a message to the handler responsible for the window indicating what action occurred. It is up to the application programmer to respond to these messages and update the window appropriately.

The API is also designed to provide device independent input/output. This is done through the device context, which contains information about the I/O device's metrics (size, etc.), as well as pen and brush settings. By using a device context, an application can write to a printer as easily as to a screen. The same formatting routines can be used for printing output and drawing on the display. In fact, the device context allows windows applications to be installed on a wide variety of platforms without the application code requiring detailed hardware knowledge.

3.2.2. DDE/OLE

In an attempt to integrate distinct applications and provide data sharing, Windows 3.0 implemented an interapplication communications method known as dynamic data exchange. With DDE a user can create a word processing document and paste in the data from a spreadsheet linked directly to the spreadsheet data and application. In this way, if the user were to modify the spreadsheet later, those changes would propagate back to the word processing document the next time it was opened. In other words, a user would only have to maintain one set of the data with the computer ensuring consistency.

DDE is accomplished through a paste buffer. When the user pastes a link into a document as opposed to just a regular paste, the client application (the one being pasted into) is informed of all the selection criteria that were involved in the original cut -- the server or source application, the source file, and the elements of the source file that were cut into the paste buffer. In this manner, when the client application reopens the document it can open up a DDE dialog with the server application and find out if the data needs to be changed.

While this approach is powerful, the data passed is still in generic clipboard format and the client application is required to know how to interpret it. For example, spreadsheet data is put into the scrap buffer as a series of values delimited by tabs. All formula and links are simply copied as their resultant values. In other words, the paste buffer limits the complexity of data that can be passed as well as its interpretation.

As a result of these limitations, a more advanced object-oriented concept known as object linking and embedding (OLE) was introduced in Windows 3.1. OLE allows applications to supply OLE servers that manage the requests of other applications. As in DDE the data is stored in the client application's document. However, here the client application has no understanding of the data that it is storing. When it needs to display the object, it issues a request to the OLE server to display it in the appropriate location. When it wants to edit the embedded object, the client application issues a request for the server to pop up an edit window that allows the user to modify the object. In this manner, various types of objects from various types of applications can all be integrated into one document without each application being required to know anything about the data of the others.

For example, a user could embed a Microsoft Draw drawing in a Microsoft Word document using OLE. The user simply selects "Insert Object" from the menu and Word queries Microsoft Windows to find out what sorts of OLE servers have been registered with the system. The user selects a Microsoft Draw object from the list provided and the Microsoft Draw OLE server is called. A window is popped up that allows the user to create a drawing, and, when done, the user simply saves the drawing into the original document. When the user returns to Word, Word does not understand any of the data it received from Draw. It therefore issues a request to Draw to display the object in the area of the window that Word requires. When the user clicks on the object, the Draw window comes back and the user can modify the object.

3.2.3. Multimedia Features

Microsoft Windows 3.1 has a large suite of tools designed expressly for multimedia applications. These features are built into the Windows 3.1 API and are available to all applications run under Windows 3.1. They include tools for sound and video.

The basic sound tools record and playback digitized .WAV files for 16-bit stereo sound. There are extensions that allow applications access to MIDI synthesizer boards and equipment. Since these features are provided through the API, applications are shielded from the specific hardware and drivers, providing a level of device independence.

In addition to sound, Windows also provides limited video capability. Coupled with Microsoft Video, Microsoft's answer to Apple's Quicktime, Windows can digitize, store and play back small video images of 320x200 pixels at a rate of around 15 frames per second. The playback facilities of Video are included in Windows 3.1, so that video files can be played on any Windows platform. Several companies also sell enhancements that allow larger video windows to be displayed as well.

With these multimedia extensions, it becomes possible to embed such things as voice annotations and short animations in such items as electronic mail and documents. However, the largest shortcoming is the lack of hardware uniformity that, despite attempts at hardware independence, has kept multimedia from becoming widely used. So far, few users have sound boards or the software and hardware required to record full motion video.

3.2.4. Pen-based Features

Anticipating the potential importance of pen-based computing, Microsoft has sought to include a large number of features in NT. In fact, in the developer's toolkit, Microsoft devotes an entire volume to pen-based applications. Microsoft apparently hopes that hardware vendors will choose to use the Windows API as the front end for their computers. In this way, applications written with pen-based hooks will work on normal computers as well as those equipped with pens.

The pen extensions in Windows 3.1 include code to decipher a common set of about 40 glyphs or pen-strokes indicating certain commands. For example, a caret (^) motion of the pen means insert here while a quick circle ending in a cursive "e" means delete this. Windows also includes features to "shapify" drawn objects. It is difficult for a user to draw a perfectly aligned rectangle without help from the computer. Accordingly, Windows takes objects drawn by the user and converts them to the most geometrically similar formal object, such as a rectangle, circle, or oval.

While these features are useful for certain applications requiring the user only to select buttons and various menu options, such as logging inventory, the real impediment to pen-based computing is in hand-writing recognition. Currently, error rates are at anywhere from 1 to 5 percent of all written characters. This may seem good, but consider that an average paragraph contains hundreds of characters and therefore potentially several errors. Until these rates are substantially improved, pen-based computing will remain specialized.

3.2.5. Windows Open Services Architecture

As a further step in its efforts to provide an open API for mail, database, and licensing services in the Windows framework, Microsoft has announced Windows Open Services Architecture. WOSA establishes strict standards for applications that may make it possible some day for users to select front-end, client software independently of the back-end, server software.

So far, the components of WOSA include the Messaging API (MAPI), Open Database Connectivity (ODBC), and a Licensing API for controlling software licenses. All applications written to these specifications will be compatible. For example, a word processor than conforms to the MAPI specification would be able to send mail using any mail service that also uses MAPI.

Developers of mail, database, and licensing services will write to Microsoft's Service Provider Interface, an API for interacting with Windows applications. Each service provider must write a software driver that translates application requests to their service equivalents. A driver for a database service, for example, would respond to queries by returning records from a database.

The success of WOSA remains to be seen, and will be determined by industry acceptance. Microsoft has attempted to increase WOSA's chances of survival by adopting existing standards, such as the multivendor SQL Access Group's standards for database access. Yet there are competing standards such as the Vendor Independent Messaging specification for e-mail, from Lotus Development Corp.

One of the biggest problems for WOSA, though, is one that plagues all specifications. By forcing applications to adhere to a rigid global standard, WOSA is forced to take a least common denominator approach. For instance, ODBC is designed to ease access to any kind of database by using a version of Structured Query Language (SQL). A Windows developer should be able to read and query, at the same time, both a dBASE format file and an Oracle database using SQL commands. The developer writing the ODBC driver for dBASE files, however, will have considerably more work than the developer of the Oracle driver (which more than likely just passes on the SQL commands to the Oracle server).

4. The Technology of IBM's OS/2 Version 2.0

IBM claims that for every system call provided by Windows NT, there is a corresponding call in OS/2 2.0. While this may in fact be true, it is probably more a direct result of the previous joint development effort with Microsoft than a convergent operating system design philosophy.

Presently, the design of OS/2 2.0 is reflective of other monolithic operating systems. Unlike Windows NT, OS/2 is not designed in a modular way. The internal use of objects is not as well established and the subdivision of operating system functions is less clear. Because of this it will be much more difficult to expand or modify the operating system.

In addition, OS/2 does not use a micro-kernel model as does Windows NT. Instead, it offers a large kernel that implements much more of the application programming interface. IBM does plan to change this philosophy in the future. The next version of OS/2 is likely to incorporate the Mach 3 micro-kernel developed at Carnegie-Mellon University, much in the same manner as Windows NT uses its own Executive.

OS/2 does implement all of the features required of a full 32 bit modern operating system. It is a fully pre-emptive multi-tasking and multi-threaded operating system that schedules CPU usage on a prioritized basis. It also provides a 32-bit flat memory model, giving it a significant advantage over the MS-DOSbased 16-bit segmented memory structure. It also provides demand-paged virtual memory as well as advanced features such as dynamic data exchange and drag and drop features. OS/2 also provides complete Windows 3.0 compatibility and support for all MS-DOS applications, unlike Windows NT, which for the sake of security and integrity only supports a subset of MS-DOS applications.

Currently, OS/2 suffers from several disadvantages as compared with Windows NT. A large portion of the OS/2 kernel is implemented in processor specific assembly language, making it very difficult to port to other architectures. Also, OS/2 2.0 does not support multiprocessing in any form and therefore will not allow users the scalable performance of Windows NT's symmetric multi-processing kernel. Its distinctly less object-oriented nature makes it more difficult to modify the operating system and to support distributed computing environments. It also does not provide support for the Windows 3.1 API, which has a large following among developers, and includes such features as OLE and vector scalable fonts (TrueType).

OS/2 does, however, have a good user interface. While similar to the Windows graphical user interface, the presentation manager interface is more intuitive and sophisticated. It provides a more enticing three dimensional look and feel, and users are able to perform most application management functions with a single click of the mouse. Moreover, OS/2 provides a powerful and intuitive Workplace Shell that is based more on Apple's Macintosh interface than Microsoft Windows. The Shell provides the familiar folder metaphor while providing object-oriented drag and drop features that allow the user to manipulate data easily. The Shell is also highly customizable, allowing users to design an interface to meet their particular needs and preferences. However, several shells have been written for Microsoft Windows 3.1 that provide all of the features of the Workplace Shell as well as a few more. For example, Symantec's Norton Desktop for Windows provides all the drag and drop functionality with a variety of other features that makes the computer much easier to use.

5. Future Object-Oriented Operating Systems

"Object-oriented" is an overworked phrase that does not have a precise technical meaning. It vaguely refers to the association of operations (procedures and programs) with data; operations exist only in

relation to data, and are accessible only through data objects.

The specifics of what an object-oriented operating system is are as yet unclear. Many systems have object-oriented properties, but also have elements of conventional programming. For example, Windows NT consists of a modular design with system "objects" that pass messages between each other and perform specific operating system functions on specific data. Windows NT also treats resources as "objects" and assigns generic methods to them. However, Windows NT is also a layered program written mostly in C, and much of its function has little to do with objects.

At least for the short term, what is usually meant by the phrase object-oriented operating system is actually an object-oriented file system. There are several file systems and file system add-ons that incorporate some object features.

On the Apple Macintosh, Finder's nested-folders-and-icons view of the contents of a disk maps directly to actual directories and files. To reorganize the current view of programs and documents, the user literally reorganizes the disk. Yet the windowing and file systems deep within the Macintosh operating system collaborate to make this approach fast and effective.

Although pure UNIX, with its reliance on environment variables and hard and soft file links, and its strong integration of the file system with program execution, is not object oriented, various vendors (Sun and HP, to name two) have layered X file management utilities that provide Macintosh-like functionality.

With Windows 3.0 tied to its MS-DOS predecessor, Microsoft took a different approach. Programs and documents could be arranged as Program Manager objects. One could create pseudodocuments as program items configured to load data files. A group called "Weekly Reports" might contain "Week of February 12" which would launch WRITE.EXE and feed it C:\WEEKLY\FEB1292.WRI.

This approach has several drawbacks. First of all, a user must initially know what file contains the weekly report and configure Program Manager with that information. The process of linking applications with data is not automated. Also, after creating the object, a user can launch it but cannot search for it. Instead, a user must retreat to the File Manager, which presents a graphical view of the physical directory tree, and hunt for something like "FEB*.WRI." Even worse, there are two sets of copy/move/delete semantics. Is the desired operation affecting a Program Manager object, merely a pointer to a file, or the File Manager object, the file itself?

A few Program Manager replacements, as well as OS/2's Workplace Shell, have attempted to deal with this problem in a better way. All of them provide a fully hierarchical logical view of the file space. Nested folders, groups, and objects seem almost Macintosh-like. Yet there is always a separate file manager around because none of these shells is able to free the user from the need to navigate the file system. Further, as the logical view grows more ambitious, the logical/physical duality worsens.

OS/2's Workplace Shell's migration tool automatically captures DOS, Windows, and OS/2 applications as Workplace Shell program-file objects. But it is up to the user to create data-file objects by dragging data file icons from the Templates folder and specifying the applications that created them.

Hewlett Packard's NewWave goes further by automatically migrating both programs and documents into its environment. NewWave works hard to capture as much as possible in its object database. For example, when a user creates a TextNote object, NewWave creates an appropriately typed document object that can be dragged from folder to folder. When a user double-clicks on it, TextNote is automatically launched. Unlike Notepad, the conventional text editor supplied with Windows, a user is not presented with a standard save dialog box when saving the object. Instead, a user can rename the object using the SaveAs option, but NewWave never lets the user get at the actual file.

The object does exist as a DOS file entity with a name on the order of C:\HPNWDATA\HPOM-F002\000006B.SRD. The user is kept unaware of this, though, and is presented with a named object instead. Should the user wish to copy the object to a floppy disk, it must first be reconstituted into a DOS file.

Ultimately, though, neither NewWave nor Workplace Shell can guarantee that a user will be able to find a given document by searching the logical namespace rather than the physical one. Applications offer the File Save dialog box that invites a user to write straight to a file without the consent or knowledge of the

desktop environment. As Jon Udell of BYTE Magazine put it, "the ability to name and use files is a sacred and inalienable right" [27].

An object-oriented file system would help solve these problems. Instead of allocating named chunks of disk space to applications, the operating system would provide opaque handles to storage objects. These objects would be registered in a global hierarchy. For example, a text editor might request an object of class Text, subclass ASCII. An image editor might instead ask for an object of class Image, subclass GIF. Moreover, objects would be given long and meaningful names unrelated to their actual storage location.

Objects could be searched for by a descriptive name, or by type information, or both. With a class hierarchy, users could conduct general searches, such as over all images, or specific ones, over all GIF images. Object linking would also be simplified enormously. Links would not break when users renamed storage objects, because the underlying handles would persist. Object linking and embedding (OLE) would not be needed as the anonymity of embedded objects would be accomplished through the regular use of handle-based storage objects.

As a move toward this type of system, Microsoft has refined its specification of OLE for OLE 2.0. OLE 1.0 requires the client to pump all of an embedded object's data through a DDE pipe to the server, which does its editing and then pumps everything back. OLE 2.0 introduces a new object storage system, the "Docfile," which gives servers direct read/write access to objects embedded in client-owned documents.

A Docfile, contained within an ordinary DOS file, can itself contain a mixture of embedded or linked objects and subsidiary Docfiles. When linking to an object in a Docfile a DOS filename is no longer used. Rather, an OLE 2.0 "Moniker," an opaque object handle, is used. This evolution will form the basis of Cairo, Microsoft's object oriented file system.

So far, Microsoft has provided only vague details about Cairo, which is scheduled for shipment in 1994. It is expected to fit into the Windows NT modular structure and replace the file system and desktop. With enforced encapsulation of the file system, Cairo may utilize advanced database and network technology. This encapsulation would enable Microsoft to implement an advanced file system that might include better security and distribution of files across a network.

The UNIX community is moving ahead with its own object oriented technology. A consortium called the Object Management Group (OMG), which includes Sun, HP, and DEC, has been formed to develop standards. OMG has developed the Object Request Broker (ORB) definition, which describes a protocol for managing distributed objects. The Common ORB Architecture (CORBA) specification defines a subset of the ORB, as a guarantee that different ORB implementations will interoperate.

Sun has independently defined ToolTalk, a higher level protocol that supports distributed tool interaction. ToolTalk is a general facility; specific standardized message types (similar in concept to Microsoft's WOSA components) have additionally been specified, including ones for desktop services (such as windowing), document and media exchange, data access and manipulation (involving data bases), and CASE interoperability. Although defined by Sun, ToolTalk is available on multiple platforms.

It is envisioned that ToolTalk will provide the transition to Sun's Distributed Objects Everywhere (DOE) technology. DOE will be a complete system, including CORBA, ToolTalk, and desktop applications that use the underlying technology. Currently, ToolTalk (which predates OMG) is in the process of being modified to use Sun's CORBA implementation (the Distributed Object Management Facility) [11].

Independent of DOE, Sun has also developed the Spring object-oriented operating system. This research project, like NT, implements the underlying system in terms of objects, clients, and servers. Unlike NT, however, it appears that Sun will not directly commercialize this technology.

IBM and Apple's joint venture in Taligent is also building an object oriented system codenamed Pink. Taligent has released very few details as to what Pink will look like, but it does not appear to build on existing technology the way that Cairo will apparently build on OLE or that DOE will build on ToolTalk.

6. Comparisons and Conclusions

The competition between Windows NT, UNIX, Apple's System 7, and OS/2 for the desktop plat-

form of the future will not be resolved quickly. Each system has a well established installed base, and there is nothing compelling about any one that will cause entrenched users to change platforms immediately.

The technology of Windows NT is most like that of Unix, and NT is most directly a threat to Unix. The remainder of this section therefore focuses on those two systems.

6.1. Windows NT versus UNIX: Strengths and Weaknesses

Technologically, UNIX and NT are both modern operating systems. They both support a large virtual memory, pre-emptive multitasking, integral peer-to-peer networking, a global network name space, and an integrated GUI. Both are portable, and both support some level of compatibility with other systems (POSIX, MS-DOS, and OS/2 in the case of NT, MS-DOS via emulation in the case of UNIX). But they both have distinctive strengths and weaknesses reflective of their backgrounds.

6.1.1. The Strengths of UNIX

UNIX has supported large networked applications for some time. Its strengths reflect that.

proven high performance

UNIX has been used for workstation and mainframe computing for ten years. It has been tuned over that period to support large computational problems reasonably well.

proven networking

From its first appearance on workstations, UNIX has been a networked operating system, and has served as the development platform for network protocols and distributed file systems.

• a flexible, distributed graphical user interface

The X Windows graphical user interface has had, from the outset, a distributed client/ server architecture. This interface is also very flexible and customizable.

• constant diverse improvement

UNIX has been subject to constant technological improvement since it first appeared as a small, standalone operating system on the PDP-11. It has managed to absorb virtual memory and networking, for example, over its 20 year history, while never being effectively controlled by any one organization.

• the availability of large applications

Because of its performance, large applications are available for it, notably data base and CAD software.

• a large number of significant vendors

UNIX is currently supported by a large number of substantial corporations (including HP, IBM, DEC, and Sun).

freeware

UNIX is famous for its relatively-freeware, such as the gnu tools, TeX, and X. A considerable amount of such software has been developed in academia.

high performance hardware

UNIX oriented hardware, even for low end desktop computing, has been more sophisticated (in terms of bus and board design) than PC hardware. This advantage will persist until NT is ported to UNIX platforms or until PC hardware improves.

6.1.2. The Weaknesses of UNIX

UNIX's weaknesses are related to its strengths.

lack of uniformity

UNIX is a standard, but only up to a point. As noted above, there is a tension between providing a common operating system and providing distinctive added value. UNIX's

diversity hinders portability. As users acquire diverse hardware platforms that diversity becomes difficult to manage.

lack of ease of use

Partly because UNIX has its roots in minicomputer timesharing, and partly because UNIX was developed, at least initially, by computer programmers for computer programmers, UNIX is difficult for inexperienced users to learn, especially compared to the Macintosh and Windows.

lack of applications

Compared to the PC world, there are few small scale applications for UNIX. This may change substantially with the introduction of WABI (Windows Application Binary Interface). It provides a Windows 3.1 x86 emulation environment for UNIX platforms. Developed by SunSelect, it is intended to be a general UNIX product for legacy application support.

expense

Compared to PC software, UNIX system, development, and application software is expensive, usually by a factor of 5 to 10 (excluding unsupported freeware). This is understandable given the relative size and nature of the different markets, but now that Windows NT is technologically moving into UNIX's market, UNIX pricing is unattractive. UNIX hardware platforms are also somewhat expensive (with the exception of PCs that run x86-based UNIX), although they almost always provide high performance.

6.1.3. The Strengths of Windows NT

Because NT has not yet been generally released, its strengths and weaknesses are more speculative and theoretical than those of UNIX. Certainly the strengths of Windows, and apparently those of NT, contrast with the weaknesses of UNIX.

• a large number of applications

There are many thousands of applications currently available for Windows. Because of Microsoft's release strategy with NT (tens of thousands of beta copies have been distributed since last July), when NT is officially released many of those applications will be ported and available. Furthermore, NT can take advantage of the extensive distribution channels for PC software.

inexpensive software and hardware

Compared to UNIX applications and platforms, Windows products (and apparently those for Windows NT) are (will be) inexpensive.

• ease of use

Most Windows applications are aimed at relatively unsophisticated users (in the computer-literate UNIX sense).

• potential reliability

Given the number of copies of Windows and Windows applications that are shipped to nontechnical users, successful products must be reliable.

extensibility

NT is apparently more modular in its implementation than UNIX.

6.1.4. The Weaknesses of Windows NT

Most of the weaknesses of NT are potential, and will not be verified until after its release.

potential lack of performance

NT's modular and layered structure comes at the potential cost of performance. It also remains to be seen how well NT supports large applications, which may require tuning of the operating system.

potential networking weaknesses

NT is designed to use Microsoft's LAN Manager, a relatively weak product compared to Novell's NetWare and UNIX's ONC and DCE. NT will supposedly interoperate with these other services, probably involving extra (probably third party) products, but NT, out of the box, will not connect up to these networks. This lack of instant compatibility, combined with Microsoft's comparative lack of network expertise, could lead to difficulties.

potential complexity

Large, distributed networks are inherently considerably more complex than standalone desktop systems. UNIX has developed into a complex system partly because of that. NT may be (or become) complex (and hence difficult to use) in response to this environment.

potential unreliability

It is possible that NT will not be robust in its first release, similar to some UNIX systems (including SunOS 4.0, Solaris 2.0, and initial versions of Ultrix and AIX). If this proves to be the case it will seriously damage NT's credibility.

6.2. Observations

This subsection collects some observations about UNIX, its vendors, Windows NT, and Microsoft, which bear on the outcome of the competition.

I. UNIX is trying to unify and advance by committee. This is a difficult and time consuming strategy, and time is not on the side of the UNIX community. Also, most of UNIX's technological advancements did not, at least initially, come from committees -- Version 7, BSD, NFS, X, Mach.

II. Microsoft is making the Win32 API available for licensing. This is significant, because the Win32 technology is the key to supporting Windows applications. If the UNIX community admits the importance of those applications, stops fighting the GUI battle, and licenses the Windows technology, then it can compete on a more equal footing with NT in terms of underlying operating system capabilities. The application availability advantage of NT will be seriously reduced.

III. UNIX is essentially technology driven. Throughout its history its development has been guided by engineers (first at Bell Labs, then at Berkeley, and later most notably at Sun). UNIX was, at least initially, written by the people who also wrote the applications (and who did the system administration). The market credibility of UNIX is based in large part on its performance, portability, modifiability, and technical sophistication.

This force has been good for UNIX, keeping it current, but it has also, in combination with the marketing differentiation mentioned above, exacerbated UNIX's tendency towards variants. There are the C shell, the Bourne shell, and the Korn shell. There are several X window managers -- mwm, olwm, twm, dxwm, and vuewm, to name a few. All this leads to nonportability and user confusion.

UNIX development has historically been experimental. Now there are large corporations developing UNIX, but the fragmentation of the market precludes dominant strategies.

IV. In contrast, Microsoft is much more user oriented and market driven. Windows was written for an infinitude of unsophisticated end users and a huge group of independent software vendors. Its market credibility is based on the number of software products that are available for it, and the robustness, ease of use, and cost of those products.

V. It is easier to make a small system larger than a large system smaller, a simple system more complex than a complicated one simple. IBM had trouble moving from mainframes to minicomputers, DEC from minicomputers to workstations. Larger systems generally involve more helper wizards at the user end, allowing the systems to be relatively hostile. Larger systems also usually involve a different cost structure supporting fewer sales with greater margins. UNIX has grown from a small system to a large one. UNIX needs to become simpler and more uniform. In contrast, Windows is getting larger with NT.

VI. An inordinate number of UNIX programmers believe that MS-DOS is junk, PCs are junk, and that Microsoft makes junk. This view is probably motivated by item 2 above -- MS-DOS is technically unso-phisticated and inelegant, compared to UNIX -- and by the view that PC users are themselves technically unsophisticated.

This attitude has blinded the UNIX community to the technical characteristics of NT.

VII. On the other hand, an inordinate number of PC users believe that UNIX is junk (when exposed to it after learning Windows or the Macintosh). The number of obscure and apparently irrelevant technical details that one must master to perform relatively simple tasks is comparatively large.

VIII. The ability of Microsoft to set standards unilaterally is not without its drawbacks. There is a certain amount of unease among users with regard to Microsoft's growing power. Indicative of this are the investigation by the Federal Trade Commission and the persistent rumors about undocumented functions in Windows that only Microsoft knows about and can take advantage of. The desire to not become dependent on Microsoft will continue to be a force favoring UNIX.

6.3. Developments to Watch For

Specific developments will be important to the UNIX-NT competition. For NT they are:

Performance. Will NT perform as well as UNIX for large applications?

Networking. Will NT provide networking facilities that are as transparent and as efficient as those of UNIX?

Reliability. Will NT, a new system, be reliable, or will it be more like some first releases of versions of UNIX?

System Administration. Will NT's system administration be easy to use (easier to use than UNIX's)?

Complexity. Will NT grow in complexity to become like UNIX as it grows to service large, distributed UNIX-like applications?

Cost. As NT becomes more complex, will it become more expensive, like UNIX?

Servers. How soon will NT appear on high performance server platforms (an important UNIX market)?

Advanced technology. How effective will NT's advanced technology be (SMP and OLE/Cairo)?

For UNIX the developments to watch for are:

Compatibility. Will the UNIX vendors offer effective compatibility for Windows applications? In particular, will WABI be successful? Perhaps more significantly, will any vendors license the Win32 API (allowing them to easily track Windows API technology)?

Standards. Will the disparate UNIX vendors agree on effective standards (in particular, will COSE be successful)?

Ease of use. Will UNIX software become more shrink-wrapped, easier to install and use?

Advanced technology. How effective will UNIX's advanced technology be (SMP and CORBA in particular)?

6.4. Market Research

Market research indicates the competition is far from over. Dataquest estimates that, for 1991, UNIX sales (including both hardware systems and operating systems) totaled 1.2 million units for \$18.2 bil-

lion in revenue. Moreover, Dataquest predicts that UNIX sales will climb to \$44.7 billion on unit sales of 4.1 million by 1996 [30]. Other analysts are not quite so optimistic about the growth in UNIX sales, believing instead that Windows NT will cut into the UNIX market. However, most analysts agree that UNIX will be around for a long time for the same reasons that MS-DOS is still around -- there is such a large capital investment in UNIX software that it simply will not go away.

Predictions are more positive for Windows NT. Even though the product is still unreleased, market analyst Rikki Kirzner sees a bright future for NT: "We believe Windows NT will be the winner, because you can't stop the power of Microsoft, but UNIX will do well" [14]. Today UNIX holds about 86 percent of the workstation market. By 1996, Dataquest predicts, that share will fall to 47 percent, largely due to NT. Dataquest also predicts that by 1996 NT will control 37 percent of the desktop market, MS-DOS will retain 40 percent, and UNIX will claim 7 percent [30].

International Data Corporation estimates that almost fifteen million units of Windows 3.x will have been shipped in the United States by the end of 1992. IDC also predicts that Windows 3.x sales will climb to over 37 million units by 1995.

The Gartner Group forecasts that Windows will have 41 percent of the desktop operating system market by 1995 (up from 12 percent in 1992) [30].

What all these predictions point to is a large and ever expanding market for Microsoft Windows based software. Since Windows NT is an extension of the Microsoft Windows 3.x API and all software written for Windows 3.x will run under NT, there is a large base of software for NT. This is the most compelling force favoring Microsoft.

6.5. Porting from UNIX to Windows NT

Much UNIX functionality is or will be available under NT. Some of it, such as networking, is built in to NT. Other functionality, primarily libraries and utilities from third parties (or the public domain), will simplify porting. This other functionality includes:

• X implementations

AGE Logic, Network Computing Devices, Hummingbird Communications, JSB, and DEC have or will have X implementations for NT.

• UNIX shells

Hamilton Laboratories, MKS, and Congruent will supply various UNIX shells and utilities, such as the C, Korn, and Bourne shells.

• gnu software

NT gnu software is available, including emacs, yacc, and lex.

• TCP/IP daemons

Such daemons as the telnet and ftp daemons are available now in the public domain. They will be part of NT when it is formally released.

NFS services

SunSelect and FTP Software will provide NT NFS client software this year.

The existence of these packages means that only system calls will have to be changed, and that the NT platforms used for porting can be integrated into a UNIX dominated network.

In addition, a third party is working on a full SVR4 UNIX API for NT.

7. Acknowledgments

We would like to thank a number of people for reading and commenting on drafts of this report: Randy Enger, Steve Gadol, John Hoch, Rick McGeer, Ellen Peel, Jon Pincus, and Brian Richards.

8. References

- [1] Doug Barney; "Plan in the works to put Windows on UNIX"; InfoWorld, November 16, 1992, p. 8.
- [2] Jonathan Cassell; "USL to Provide Technology to Fight Microsoft Windows NT"; *Electronic News*, September 7, 1992, p. 22.
- [3] Kaare Christian; The UNIX Operating System; John Wiley & Sons, New York, 1988.
- [4] Peter Coffee; "Win NT, OS/2 2.0 and Solaris 2.0 vie for Honors as Top 32-Bit OS"; *PC Week*, August 17, 1992, p. S8.
- [5] Ron Copeland; "Microsoft takes Windows on the road to Cairo"; InfoWorld, July 13, 1992, p. 8.
- [6] Helen Custer; Inside Windows NT; Microsoft Press, Redmond, Washington, 1993.
- [7] Rik Farrow and Frank Hayes; "The PC UNIX Explosion"; UNIXWORLD, October 1992, pp. 48-52.
- [8] Mary K. Flynn and Robin Raskin; "Windows Starts a Family. But are there too many members?"; *PC Magazine*, September 29, 1992, pp. 29-32.
- [9] Frank Hayes; "Going Graphical"; UNIXWORLD, July 1992, pp. 46-52.
- [10] Frank Hayes and Gary Poole; "Novell to the Rescue"; UNIXWORLD; March 1993, pp. 42-48.
- [11] Astrid Julienne and Larry Russell; "Why You Need ToolTalk"; SunExpert; March 1993, pp. 50-58.
- [12] Stephen Manes and Paul Andrews; Gates; Doubleday, New York, 1993.
- [13] Vance McCarthy and Cate Corcoran; "Univel targets UNIXware release for early fall"; *InfoWorld*, May 11, 1992, p. 1.
- [14] Gary Andrew Poole; "Sun's Soft Spot"; UNIXWORLD, November 1992, pp. 52-57.
- [15] Tom Quinlan; "IBM still tickled Pink about Taligent operating system"; *InfoWorld*, August 31, 1992, p. 8.
- [16] Omri Serlin; "USL's Battle for the Desktop Market"; UNIXWORLD, October 1992, pp. 39-40.
- [17] Paul M. Sherer; "Microsoft Bets Its Future on NT"; PC Week, August 17, 1992, p. S1.
- [18] Paul M. Sherer; "Windows NT Gets Mixed Reaction"; PC Week, September 28, 1992, p. 4.
- [19] Paul M. Sherer; "Windows for Workgroups Unites 2 Fronts"; PC Week, August 24, 1992, p. 1.
- [20] Paul M. Sherer; "Microsoft: Win 3.1 Sales Exceed 1M per Month"; PC Week, August 24, 1992, p. 1.
- Jim Seymour; "OS/2 Still Suffers from IBM's Lack of Foresight"; PC Week, September 28, 1992, p. 75.
- [22] Eamonn Sullivan and Larry Seltzer; "Microsoft's Cairo May Bring Users to New Heights"; *PC Week*, August 17, 1992, p. S/20.
- [23] Eamonn Sullivan; "Developers Get Clean Windows With Open Services Architecture"; *PC Week*, August 17, 1992, p. S/20.
- [24] Michael Sullivan; "USL's Destiny: UNIX for the masses"; Computerworld, November 9, 1992, p. 40.
- [25] Kevin Streho; "Microsoft makes its move with Windows NT SDK"; InfoWorld, July 13, 1992, p. 1.
- [26] Dave Trowbridge; "USL Gets Firm Hand On The UNIX Throttle"; Computer Technology Review, February 1993, pp. 1,6.
- [27] Jon Udell; "In Search of an Object-Oriented File System"; BYTE, September 1992, pp. 307-308.
- [28] Mitch Wagner; "NT Being Shored Up With Server Version"; Open Systems Today; March 1, 1993, pp. 1,69.
- [29] Mitch Wagner; "Vendors Ally Against NT"; UniForum Daily; March 19, 1993, pp. 1,30.
- [30] Tom Yager and Ben Smith; "Is UNIX Dead?"; BYTE, September 1992, pp. 134-145.
- [31] IEEE Standard Portable Operations System Interface for Computer Environments; IEEE Std 1003.1-1988; 1988.

The Status of Windows NT and Its Competitors

William R. Bush Brad Peters A. Richard Newton

30 December 1993

Department of Electrical Engineering and Computer Sciences University of California Berkeley, California 94720

Executive Summary

With its new platform, Windows NT, Microsoft has caught up technically with Unix. So far, however, the Unix market has not been noticeably affected.

There are very few native NT applications. It is anticipated that important applications from Microsoft and others will appear in the first half of 1994.

In the meantime, sales of NT have been moderate (large for the Unix market, small for the PC market), estimated at between 200,000 and 250,000 for the first three months. Sales have dropped recently, indicating that early adopters are waiting for applications.

NT has been getting misleading negative coverage from the PC press; some of its members have misperceived NT as a replacement for Windows 3.1 and are disappointed with its resource requirements.

Long-term projections for NT in 1995 and 1996 are positive, predicting that NT will spread into the Unix and Windows markets.

The Unix vendors face several long-term problems in competition with NT: uniformity and portability, cost and price, and ease of use. They have strategies for partial uniformity and portability, which will be implemented in 1994. All the Unix hardware vendors have or are developing NT platforms.

Porting 32-bit C programs to NT is not particularly difficult. NT looks like a mutant Unix with a different GUI. A number of tools exist to ease the process.

IBM's OS/2 is emerging as a serious competitor to NT, technically sophisticated (almost on a par with NT and Unix), mature, and fairly well established among corporate PC users.

Recommendations: there is not an immediate need to move to NT, but it should be investigated and planned for; the appearance of NT applications and the success of OS/2 should be closely monitored.

1. The Status of NT in Quotes

"If you don't know why you should need NT, then you don't need NT." -- Bill Gates, Microsoft [1]

"NT just doesn't bring capabilities to mainstream desktops that justify its additional hardware and software costs."

-- Larry J. Seltzer, PC Week [2]

"As for NT, the operating system has fallen from grace as the next generation Windows. Despite Microsoft's marketing muscle, NT sales will never approach those of Windows 3.1 and its successor, Chicago."

-- Carter Lusher, Gartner Group [3]

"NT is a real operating system... NT is a Windows that even Unix folks could love." -- Carl Dichter, SunWorld [4]

"Let's face it, Windows has won the desktop, and the solutions proposed for running Windows apps atop Unix haven't lived up to their promises. Sure, there always will be Unix desktops floating around, but corporate America has voted for Windows and, as Walter Cronkite would say, 'That's the way it is.'" -- Sally Atkins, Open Systems Today [5]

"It's clear that the hardware industry is falling all over itself to support NT." -- Larry J. Seltzer, PC Week [2]

"Nothing is more critical to the IBM company than OS/2 and AIX." -- Louis Gerstner, IBM [6]

"The one thing we can't risk is bringing Apple to its knees by open licensing the OS and trying to turn to a software-only business."

-- Michael Spindler, Apple Computer [7]

"Many market watchers, including International Data Corp. and Dataquest Inc., predict Windows NT will be outselling other high-end operating systems in three years."

-- Don Clark, Wall Street Journal [8]

Windows NT sales for its first three months are estimated at 200,000 to 250,000 [9]. Microsoft estimates the current Windows 3.1 installed base at 40,000,000 [10], and that Chicago sales will be 10,000,000 in the first three months of release [9]. MS-DOS 6.0 sales were 6,000,000 in its first 90 days of release [8]. IBM estimates the OS/2 installed base at 4,000,000 (Microsoft estimates 2,500,000) [11]. The MS-DOS installed base at the end of 1992 is estimated at 51,000,000, and the 1992 Unix installed base at 4,000,000 [12]. Sun Microsystems shipped more than 250,000 systems this last year, bringing its installed base to over 1,000,000 [13]. Intel estimates shipping up to 10,000,000 Pentium chips in 1994, 15% of the worldwide computer market [14]. Mark LaRow, Ernst & Young [12]: "It doesn't matter whether you call it open, proprietary, international, or de facto. There's no such thing as a real standard. What matters is market share."

2. Introduction

This document reviews the status of Microsoft's Windows NT and its competitors, as of December 1993. The following section summarizes NT's current status; sections at the end discuss trends and possible developments. Sections in between review:

- results of NT benchmark tests,
- the state of the NT market,
- current NT products,
- future Windows products,
- · software development for NT,
- and competition from IBM's OS/2, from Unix, and from Apple.

3. Summary: Overview Status of NT

Windows NT is Microsoft's long-term platform of the future -- its "core technology" [15]. With the current version of NT Microsoft is beginning to move applications and users to that platform. As Microsoft's Senior Vice President Paul Maritz has said [15], "We are laying down the technological foundations for three to four years from now. People can start deploying it in the next six to 12 months."

As with any new system, that movement will take some time. NT is immature in terms of native applications; essentially all it does now is run Windows 3.1 applications, albeit with a much more robust underlying operating system. "Most Windows 3.1 systems are probably best left as is until that first killer NT application comes along" [16].

NT's ability to run Windows 3.1 applications gives it credibility, but hasn't helped it much in the marketplace. Most importantly, most installed PCs do not have the hardware resources that NT needs. As *PC Week* observes [2], "NT just doesn't bring capabilities to mainstream desktops that justify its additional hardware and software costs."

NT sales, estimated at between 200,00 and 250,000 for the first three months of release, reflect this reality. In contrast, Microsoft sold 6,000,000 copies of MS-DOS 6.0 in 90 days [8].

Near-term, for most existing desktop systems, the upcoming upgrade of Windows 3.1 is much more important. That product is Windows 4.0, known as Chicago, due out in the second half of 1994. Its primary anticipated benefits are a new, more Mac-like graphical user interface, and an implementation that takes advantage of the hardware features of the Intel 386 processor, resulting in greater performance and reliability than Windows 3.1. Microsoft estimates that sales of Chicago in its first three months will be 10,000,000 [9], and up to 50,000,000 in the first year [17]. Windows 3.1 has a current installed base of 40,000,000, which could grow to 55,000,000 to 60,000,000 by the time Chicago ships [18].

It appears that the porting of 16-bit Windows applications to 32-bit NT will take some time. Robert Lee, a Lotus manager, says that [19] "[Porting a 16-bit Windows application to 32-bit Windows NT] is full of minor pitfalls, but is not technologically challenging." The problem is that the optimizations previously performed to exploit a 16-bit [Windows 3.1] structure are lost. "And almost all applications are pretty seriously optimized." Porting of 32-bit Unix applications may in fact go more quickly [4].

Despite the current lack of applications and hesitant market reaction, "the hardware industry is falling all over itself to support NT" [2], sensing an opportunity to get a piece of Intel's market. The utility of these alternative hardware platforms remains to be demonstrated, however, since it is unclear how many applications will be ported to them.

The Unix community also appears to be taking NT seriously, which is not surprising since there is a large overlap between computer hardware vendors and Unix vendors. The response has taken the usual Unix form: standards. In an attempt to provide a unified opposition, a standard set of system calls and utilities is being specified; implementations of this standard will be branded "Unix" [20]. Scepticism and ennui abound -- a headline in *UnixWorld* [21] reads "Unix Vendors Sing the Same Tired Tune (Again)."

Whatever the outcome of the standardization efforts, the widespread belief is that Unix will be a long-lived niche product. As David Smith, an IDC market analyst, says, "The likely impact of NT is not that it will displace Unix, but rather that it will slam the door on Unix's access to the PC upgrade market" [13]. It has lost the desktop but will continue as an important server platform.

IBM, capitalizing on disappointment in NT, is pushing OS/2, vowing "to build OS/2 into the world's operating system of choice by 1995" [11]. It appears that OS/2 will make inroads into the Windows market, but it remains to be seen how extensive they will be. OS/2 is strongest in Fortune 1000 companies, IBM's traditional customer base [22].

In a survey of MIS managers conducted by Computerworld, NT did relatively well, compared to OS/2 and Unix [23].

OS Characteristic	NT	OS/2	Unix
interoperability	3.7	2.8	3.0
features	3.7	3.2	3.5
user friendliness	3.7	3.1	2.1
manageability	3.6	3.4	2.6
performance	3.2	3.3	3.9
cost-effectiveness	3.2	3.4	2.8
service and support	3.0	2.7	2.8

Note that NT scored better than the others in all categories except for performance (3.2 to 3.9 for Unix) and cost-effectiveness (3.2, versus 3.4 for OS/2); the worst score was Unix's user friendliness (2.1, versus 3.7 for NT), and the next worst was Unix's manageability (2.6, versus 3.6 for NT).

Users appear willing to be patient towards NT, at least for the moment. In a survey of 1000 computer professionals NT was the number one purchase projected for 1995 and 1996 [8]. And the *Wall Street Journal* reports that IDC and Dataquest predict NT will be the sales leader in "high-end operating systems" in three years. The Journal did not define "high-end"; Unix and OS/2 are considered high-end by most PC users (at least in terms of resource requirements), while MS-DOS, Windows 3.1, and Chicago are not.

Nicholas Baran of *PC World* summarizes the current state of NT for users considering moving to it with the following if-thens [24]:

- If you use a standalone PC with ordinary Windows applications, then you should wait for Windows 4 (or go to OS/2).
- If you run ordinary business applications, then you should wait a year and reevaluate.
- If you run specialized applications requiring a lot of power, then you should consider upgrading to NT when the applications you run are available.
- If you run workstation-class applications on UNIX or minicomputers, then upgrade to NT as soon as the applications are ported.

4. Performance and Reliability of NT

Initial experience with NT shows it to have acceptable performance and reliability.

In particular, in comparison with Windows 3.1, running 3.1 applications, performance has been assessed as "much slower" to "somewhat faster", and in general described as "more than adequate" [2]. Performance on 386 with 12 megabytes of memory was "tolerable" but not recommended [16]. An "impressive improvement in speed" was noted between the March beta version and the final release [25].

Tests done by *BYTE Magazine* on assorted 486 and Pentium platforms indicate that, in general, Windows applications run one half to three quarters as fast under NT as under Windows 3.1 [26]. The following table (from page 92) presents the results (with 1.0 being the performance of Windows 3.1):

Application	Platform 1	Platform 2	Platform 3	Platform 4
overall	0.60	0.56	0.66	0.78
word processing	0.49	0.31	0.53	0.62
spreadsheets	0.79	0.71	0.73	0.76
database	0.76	0.84	0.76	0.96
graphics	0.64	0.44	0.39	0.38
file I/O	2.20	2.90	1.45	2.33

Platform 1 is a Unisys Pentium/60; Platform 2 is a Gateway 486DX2/66; Platform 3 is an Everex Step 486DX2/50; Platform 4 is an IBM PS/2 486DX/33.

Application	Platform 1	Platform 2	Platform 3	Platform 4
overall	0.82	0.66	0.64	0.85
word processing	0.73	0.48	0.47	0.67
spreadsheets	1.00	0.86	0.79	0.96
database	0.85	0.77	0.70	0.92
graphics	0.96	0.72	0.59	0.78
file I/O	0.74	0.99	0.90	1.10

In contrast, OS/2 is noticeably faster, but still noticeably slower than Windows 3.1:

Note NT's performance with (native) file I/O. A Microsoft spokesman has indicated that native Win32s prototype applications are 1 to 3 times as fast as their Windows 3.1 versions [27], which is possible given the file I/O results.

BYTE also compared NT on different platforms. On average, the Pentium was 3 times the speed of the IBM PS/2, the SGI/MIPS Magnum R4400/75-150 5 times, and the DECpc Alpha/150 6 times. In detail (from page 94):

Platform	numsort	simple FPU	bit-field ops	strsort
IBM PS/2	1.00	1.00	1.00	1.00
Everex	1.45	1.11	1.49	1.45
Gateway	1.96	1.80	1.71	1.06
ALR Flyer	2.13	1.88	1.74	1.45
Unisys	3.21	3.31	3.18	3.13
SGI/MIPS	4.18	5.19	5.16	6.61
DECpc	4.64	4.86	6.90	9.17

4 600

.

(The ALR Flyer is a 486DX2/66.)

Other experience indicates that emulated applications run at 486 speed on an AXP, and Excel on an R4400 runs twice as fast as on a Pentium.

PC Week tested Microsoft's Visual C++ product under Windows 3.1 and NT and, for one test program, noted the following results [25]:

version	compile time (seconds)	code size (bytes)
3.1 debug	40	1.2 M
NT debug	65	2.3 M
3.1 release	38	83 K
NT release	47	108 K

PC Week also tested Microsoft's SQL Server for Windows NT, a data base server that takes advantage of symmetric multiprocessing and has an 8 terabyte capacity [28]. It yielded the highest transactions per second measurements *PC Week* has observed, performing as well as some (unspecified) multiprocessor Unix database servers [16]. The specific test was the RiteSize II benchmark, tested with 1 to 60 workstations submitting queries and with 1, 2, and 4 CPUs on a Sequent WinServer 3000 with 50 megaHertz 486 processors.

Number of Workstations	4	8	12	16	24	36	48	60
1 CPU	23	82	88	83	82	78	75	70
2 CPUs	24	75	115	126	129	128	125	122
4 CPUs	25	75	121	135	136	140	137	133

SMP performance, due to Windows NT, was effective, with a 69% throughput improvement with 2 CPUs and a 85% improvement with 4. *PC Week* also observed that: "Microsoft has done an outstanding job with the database management tools", which are GUI-based; "ease of networking, as a general rule, seems to be one of NT's greatest strengths" [16]; and NT's administration and security tools were very useful.

Other experience with Windows NT SMP performance appears positive [29]. One company moved from two OS/2 servers to a Sequent NT WinServer for production line control, for the benefits of a single server and NT's administration tools. A 30% improvement in line scheduling performance was observed, as well as a marked increase in response time to data base queries (up to 30 times faster). Unix was rejected as unfamiliar and expensive.

Reliability has not been a problem with NT, or at least has not been mentioned. Nonetheless, Microsoft released a set of 96 bug fixes in October [30]. Most of the bugs were either high-stress (involving, for example, 14 disks and 400 processes, or 4 gigabyte files) or configuration specific (for particular drivers).

Our general experience with NT has been positive. We have installed the beta and final releases on a DECpc, and the final release on a DEC AXP 150. The process was straightforward. System administration has been easy; the user is completely shielded from magic system files. The final release of NT has not crashed.

5. The NT Market

NT sales have been moderate, and applications slow to appear, but long-term projections remain positive.

5.1. NT Sales

According to NT product manager Dwayne Walker, Microsoft's primary target NT market (at least initially) is 2,000,000 to 5,000,000 corporate "power users" [31]. At the NT product announcement, Bill Gates set a sales goal of 1,000,000 in the first year [1], roughly the size of Sun's installed base [13].

It is estimated that 100,000 beta versions of NT were shipped [15], but that sales of the final release are slow. Two independent estimates put sales at between 200,000 and 250,000 ([32], [9]). Brent Williams of IDC says this is merely the "initial stages of rolling out an operating system" and predicts sales of 400,000 units by the end of 1993 [15].

In the "Software Saleswire" feature of *PC Computing* ([33], [34], [35], [36]), compiled from data from 22 distributors, NT appeared in the top 10 in October, finishing 7th (in the "Utilities" category). This surge probably reflects interest by developers and experimentation by early adopters. In November it had fallen to 15th. During the same period Windows 3.1 was first, MS-DOS second, and OS/2 2.1 fourth. The complete results:

Product	Company	July	August	September	October	November
Windows 3.1	Microsoft	2	2	1	1	1
MS-DOS 6.0	Microsoft	1	1	2	2	3
QEMM-386 7.0	Quarterdeck	4	4	4	3	2
OS/2 2.1	IBM	-	3	3	4	4
Norton Utilities 7.1	Symantec	5	6	5	5	6
After Dark 2.0	Berkeley Systems	3	5	6	6	5
Windows NT	Microsoft	-	-	-	7	15
Saber LAN Workstations	Saber Soft- ware	-	-	10	8	9
More After Dark 2.0	Berkeley Systems	-	-	8	9	-
Visual Basic 3.0	Microsoft	-	-	-	10	-
Hijack Pro 2.0 for Windows	Inset Sys- tems	-	8	9	11	10
Windows for Workgroups	Microsoft	6	9	-	13	7
CA-Clipper 5.1	Computer Associates	9	10	-	17	8

A "-" indicates an unspecified ranking greater than 10.

5.2. NT Resource Requirements and Features

At the official announcement of Windows NT Bill Gates said [1], "If you don't know why you should need NT, then you don't need NT." As the article reporting the event remarked, the above suggests "just how far NT is from Microsoft's traditional stronghold among low-end commercial users."

Specifically, Microsoft itself says, "If you're just running personal productivity apps, then Windows 3.1 is the right platform" [27]. (Personal productivity applications are word processors, spreadsheets, and the like, which require relatively few computing resources.) In contrast, NT "has many features that would become extra baggage to most Windows 3.1 users, such as security and built-in networking" [24].

On the other hand, "NT is a platform for any type of client/server solution, and it's also for users who want additional power and security" [15]. Rikki Kirzner, of Dataquest, observes [27]: "The user's question is, Do I need more power, memory, or multitasking? If I do, then I need more than DOS and Windows, and my choices are NT and UNIX. If I want seamless integration, more or less, then NT is my choice." (Note that OS/2 is not mentioned.)

So far, NT appears to be successful with the right kind of users. In its best-of-the-year salute to NT, *PC Week* said [37], "Buyers who had kept Windows NT in proper perspective were not disappointed with it."

Unfortunately for Microsoft, the publicity surrounding NT led the broad base of PC users, who are primarily market followers, to see it as a general replacement for Windows. As information about NT's resource requirements, lack of new user-visible features, and lack of applications spread, these users have become critical, and enthusiasm has cooled. Microsoft admits that it has not managed user perception well. Jonathan Lazarus, a vice president at Microsoft, has said, "We've been trying to set people's expectations, and we failed" [8].

Carter Lusher, of the Gartner Group (a market research firm) says that NT "has fallen from grace as the next generation Windows. Despite Microsoft's marketing muscle, NT sales will never approach those of Windows 3.1 and its successor, Chicago" [3]. Another analyst observes that "most clients are just barely settling into Windows 3.1. In terms of volume, I don't expect NT to be an important factor in the marketplace for at least two years" [24].

The fact is that NT realistically requires at least a 486 with 16 megabytes of memory and a 200 megabyte disk [24], which makes it unusable for many installed PCs. Hence the remark that "recent comments on NT seem aimed at declaring it dead at birth from a heretofore unknown disease: infantile obesity" [38].

Further, it is clear that the "hidden" technical features of NT -- multitasking, virtual memory, and object oriented RPC-based implementation -- are not valued by most (market follower) PC users, even though they contribute to a considerably more reliable system than MS-DOS. One reviewer, in fact, had it backwards when she observed, "Windows NT has the flash, but it needs more substance" [39]. Presumably by "flash" she meant "visibility". In reality, NT has as much technical substance as, if not more than, other "desktop" operating systems (DOS/Windows, OS/2, and Mac OS), but it does not have many new user-visible features, and none that are glamorous.

5.3. (Im)Maturity

Even though NT runs all (16-bit) Windows applications, and is reasonably robust (it appears as solid as Solaris 2, for example [40]), it is still an immature system. Very few 32-bit applications for it exist, substantial improvements in it are scheduled for the next release [41], and it has simply not been around long enough for users to be familiar with it and confident in it.

In other words, early adopters are still waiting for applications and improvements, and market followers will not use it for some time. As Sam Whitmore of *PC Week* writes, "Selling an NT software development toolkit to a code jockey or conscripting vertically integrated also-rans such as NCR and Intergraph is one thing. Training a newly certified VAR to convince Joe Doakes at Aetna that everything MVS or Unix can do, NT can do better, is quite another" [42].

The most important factor in overcoming the "NT stands for Nice Try... Not There" [7] attitude is the development of native 32-bit applications. Fortunately for Microsoft, it produces some very popular personal productivity applications, and is porting them to NT.

This is significant because, despite Microsoft's promotion of NT as a server platform, users want it to be a desktop platform as well (and "native NT productivity applications for the desktop, which can influence server usage, are virtually nonexistent" [15]). A ZiffNet poll with 72 respondents [43] showed that most planned on evaluating NT either as a desktop only system (58%) or server and desktop (38%). A man-

ager at Quick America, a developer of financial trading software, notes that [15] "users would feel more comfortable with NT, even if they're using it as a trader workstation, if the major desktop applications were available for it."

The response of vendors is mixed so far (see [27]). Banyan, for example, has put its release of an NT version of its Enterprise Network Services on hold because "acceptance of NT is not yet sufficient to justify the work on Banyan's part" [44]. And WordPerfect has said, "If NT becomes a major platform, we will be there to support it" [27]. On the other hand, HP "cannot ignore the existence of NT or its gradual evolution, nor can our partners" [45].

Independent of applications, a major aspect of user confidence is support, particularly with a new system. A Gartner Group survey [46] shows that support is 45% of the cost of software, and the most recent J.D. Power Desktop Computer Satisfaction Survey [47] shows that the most important factor in customer satisfaction is support. Microsoft has moved to address this issue by enlisting DEC to provide NT support [48]. Rober Herwick, of Hambrecht and Quist, says that "historically Microsoft has a horrible reputation for support -- DEC has to stand behind it to be credible."

5.4. Near, Medium, and Long Term

In the near term, while NT is still relatively untried and few applications exist for it, early adopters and analysts appear patient. An IS executive with a large oil company noted that "there are still some pieces missing, but as long as they come together in mid-1994, that's fine. The second release will be the safe release" [9]. This second release is an update planned for mid-1994, codenamed Daytona [49]. And Microsoft is reassuring users; James Allchin, vice president of advanced systems, says, "We've got the largest product group at Microsoft working on making NT dramatically better. We are heavily committed to this sucker" [8].

Microsoft probably has about a year to develop the critical mass of credibility needed to draw in the market followers. Michael Dell, of Dell Computer, however, warns, "It's going to take longer than most people think" [8].

Analysts are positive about NT in the medium term, believing the market followers will appear. They assume Microsoft will be able to build that critical mass and make NT a significant high-end product. "Many market watchers, including International Data Corp and Dataquest Inc, predict Windows NT will be outselling other high-end operating systems in three years" [8]. The *Wall Street Journal* says, "Windows NT: off to a slow start, but could become a corporate standard in two or three years" [50]. In a survey of "1000 computer systems professionals", NT was the number one purchase projected for 1995 and 1996 [8]. And another survey of IS managers showed that one of "biggest winners" for 1996 will be NT (big losers will include MS-DOS, Windows, and the Macintosh) [12]. One analyst, however, predicts moderate success (from the PC perspective): "I think the NT market is going to be a couple percent of the PC market. I think a wild success scenario for NT at this point would be 5% or 10% -- ten percent's probably almost inconceivable" [51]. Note that 5% of the installed base of Windows is 2,000,000 units.

In the long term, "there should come a point somewhere in the second half of the decade where the additional resource requirements of NT are not a problem" [51], and it can become the mainstream Windows platform. That will not be the NT of today, however. By then the object oriented Cairo system will have been released (along with competing technologies from other vendors, including Taligent and Sun), and NT will track the Cairo technology. It appears that NT will be the basis of whatever long-term platform Microsoft develops.

6. NT Products

Although NT is immature, products for it are starting to appear. Still, NT made *PC Week*'s list of disappointments of 1993 because "very little application software, server-based or desktop, has surfaced for the new operating system" [52].

6.1. NT Software Products

There is some uncertainty about how many 32-bit NT applications exist. Microsoft said that 125

Intel applications existed at the end of October [53], and approximately 225 applications were due out by the end of November [32] (most of them full Win32 applications, not Win32s), with 500 anticipated by year's end [15]. At Comdex 180 NT developers were scheduled to show applications [54]. *PC Week*, however, observed that "MS claims that there are more than 200 applications currently available for NT, yet many ISVs in the Microsoft booth [at Comdex] were showing unfinished NT applications" [9]. For non-Intel platforms, as of November, MIPS says there are 200 for the MIPS and DEC says there are 127 for the Alpha, but Carrera, a NT hardware systems vendor, says there are 20 for the MIPS and 16 for the Alpha [53]. For comparison, 8,000 applications exist for SunOS, 1300 for Solaris 2, and 500 for Solaris 2 on the X86 [55]. These counts are only suggestive, saying nothing about the quality or utility of the applications; individual mileage may vary. Notably, some Unix products such as FrameMaker, Oracle, and Informix are appearing as soon as comparable PC products [32].

The following subsections discuss products described in articles. In addition, terse listings of NT products, present and future, can be found in [27] and [19].

6.1.1. NT Software: Personal Productivity

PC desktop products are due out in 1994, including Word [15], Excel [15], WordPerfect [15], PowerPoint [27], and Lotus 1-2-3 [27]. These products are prominent on *PC Computing*'s November 1993 Software Saleswire for business applications [34].

November Ranking	Product	Company	Projected NT Availability
1	WordPerfect 6.0 (DOS-based)	WordPerfect	-
2	Excel 4.0 for Windows	Microsoft	Q1 1994
3	WordPerfect 5.2 for Windows	WordPerfect	Q2 1994
4	Lotus 1-2-3 4.0 for Windows	Lotus	1994
5	Access 1.1 for Windows	Microsoft	-
6	Word 2.0 for Windows	Microsoft	Q1 1994
7	Publisher 2.0 for Windows	Microsoft	•
. 8	Quattro Pro 5.0 for Windows	Borland	-
9	Winfax Pro 3.0 for Windows	Delrina	-
10	Procomm Plus 2.01	Datastorm	-

6.1.2. NT Software: Networks

Microsoft has released a beta version of its Novell network connection software [56]. Novell has released a pre-beta NetWare requester, which is missing features and is "sluggish" [57], and does not use NT's built-in NDIS network interface (but rather its own ODI). Problems with these products continue to be reported [58]. Beame & Whiteside has announced software to let Novell clients access files and printers on NT machines without a NetWare server [59]. These products are important, because Novell networks comprise 70% of the PC networking market [39].

Microsoft has announced an NT version of SNA, for connections to IBM mainframes [60].

IBM, Microsoft, and DEC have announced collaborating on products for enterprise network administration [61]. Specifically, IBM and Microsoft both have administration products (NetView/6000 SNMP and Hermes, respectively). They will be modified so they can share information and requests. An NT port by DEC of NetView/6000 is scheduled for release in the second half of 1994.

Separately, HP has announced a port of its OpenView network management platform to NT, with links to Microsoft's Hermes, available in 1994, and Cabletron will port its Spectrum for Open Systems to NT [62].

Beame & Whiteside have announced an NT NFS product for file sharing with Unix systems, available in the first quarter of 1994 [59]. And FTP Software has demonstrated an NT NFS product but has not announced it [59].

In addition DEC will provide DCE and ObjectBroker support [32].

6.1.3. NT Software: Databases

Microsoft is now shipping its SQL Server for Windows NT [28] (it costs \$995 single user, \$2995 for a workgroup of up to 10 users, and \$14995 unlimited). It is readying Alpha and MIPS versions [63]. The product uses the Advanced Server version of NT, which supports a transaction-based file server with RAID level 5 [64].

Watcom has announced an SQL server for NT (as well as OS/2 and NetWare) [65].

Informix has made a deal with Microsoft to release a data base server bundled with NT, thereby placing NT in the OEM market [56]. The Informix-SE low-end server is to appear in January [66]; bundled with NT it will cost \$1795. An Informix Vice President of marketing notes that there is "a lot of interest among VARs in exploring NT as an alternative to Unix."

6.1.4. NT Software: Compilers

Microsoft is shipping its Visual C++ environment for NT [25]. It currently supports only the Pentium. Versions for other platforms will appear in 1994.

Microsoft is also shipping a FORTRAN development environment for NT (the FORTRAN Power-Station 32), which includes a compiler, integrated debugger, browser, and 32-bit graphics libraries [67]. It is aimed at mainframe and minicomputer users -- it supports MVS and VMS FORTRAN dialects.

Borland is shipping its C++ environment, which includes a graphical debugger and object windows library. It supports both the Win32 and Win32s APIs.

Symantec is shipping its 32-bit C++ compiler, which includes the Microsoft Foundation Class (MFC) library found in Visual C++ [68].

6.1.5. NT Software: Et cetera

Microsoft now provides an NT Resource Kit containing various utilities, including ones for network management and performance monitoring, as well as a Korn shell and a compiler front end [69].

Also being ported to NT are Silicon Graphics' OpenGL 3D graphics, X.500, and the Taligent object framework (the last by IBM) [32].

NCR sells a line of Intel MP servers, and it has announced a porting program for its corporate users to migrate them from OS/2 to NT [70]. It is also porting its in-house software to NT.

6.2. NT Hardware Products

The state of NT hardware products is more definite than that of software. There are several NT RISC platforms available, with more on the way. It appears that Bill Gates' goal of NT on every major RISC processor by 1995 [31] will be met. Microprocessor vendors have embraced NT as a way to get a piece of Intel's market. "The hardware industry is falling all over itself to support NT" [2]. Put another way, one effect of NT has been "to legitimize the non-Intel architectures" [2], and RISC in general.

Phil Hester, a Vice President at IBM, asserts that there is a fundamental 2x RISC price/performance advantage over Intel [71]. Michael Slater, the publisher of the *Microprocessor Report*, says RISC has "probably better than a two to one advantage today" overall, and a larger advantage with floating point specifically [51]. One Alpha user reports "very significant improvements" [72].

However, Slater does not see a big move to RISC processors in the near future. "Being 30-50% slower than the fastest RISC processor is not going to be enough to get people to put up with an enormous software transition. The vast majority of the market is not at the leading edge of performance" but "a few years behind" [51]. Thus the market for each RISC architecture is not going to be millions of units, but it "could still be a significant opportunity within the scale of the volumes they're at today." "There should come a point somewhere in the second half of the decade where the additional resource requirements of NT are not a problem", and software ports to NT are routine, and the "hardware infrastructure" (support chips and boards) is in place, so that NT RISC platforms could become significant in the market. In particular, Slater sees an opportunity for Motorola's PowerPC to become a "high-volume standard" in the 1996-1997 time frame.

Current RISC architectures with NT ports are the DEC Alpha and the MIPS, both with low-cost integrated chipsets announced ([73], [74], [75]), and one MIPS NT PC announced for under \$3,000 [76]. High-performance servers have also been announced for both ([77], [72]), and a color Alpha notebook and a 275 MHz server have been shown [72]. Vendors include Carrera Computer, NeTpower, DeskStation Technologies, DEC, and NEC [53].

A PowerPC version has been shown [9] with an NT port by Motorola and IBM [78]. Microsoft will distribute the port to other vendors and will sell a shrink-wrapped version [79]. Appearing later will be a new SPARC chip, developed by Sun and Intergraph, with an NT port ([80], [81]). And HP has announced a workstation for less than \$4,000 that will run NT (although no firm release date for the NT port has been given), signaling "the end of resistance from one of the last NT holdouts" [82]. "Industry analysts agreed that HP anticipates a shift in the workstation market once rivals Sun Microsystems and IBM field workstations that run NT next year" [45].

Other hardware products indicate that NT is being taken seriously as a server. A *PC Week* survey of Intel-based MP servers indicates that 18 of 21 systems run NT [83]. And NCR RAID products will have NT support [84].

On the network side, Cisco will make Ethernet and token ring router cards for NT, due out in the last half of 1994 [85].

Finally, the price of Pentium-based PCs continues to drop (to under \$3,000 per system [86]), which helps NT by making it easier to acquire the more powerful hardware that NT needs.

7. The Range of Windows Products

The Windows product line continues to evolve, both NT and its less ambitious, non portable, more common relatives.

7.1. Windows NT 3.11 -- Daytona

Microsoft has already started discussing the next NT release ([87], [41], [18], [88]). It is due out the second quarter of 1994, and will be designated Windows NT 3.11. It is scheduled to begin beta tests in January.

The primary development goals are slimming down (but not to Windows 3.1 size) and speeding up. It will require 12 megabytes of memory (4 less than the current release), and, based on initial testing, will run up to two times faster on a server.

In addition, DoubleSpace disk compression, Microsoft's NetWare redirector, 32-bit OLE (version 2.01), and Silicon Graphic's OpenGL 3-D graphics technology will be included. (The support of NetWare represents an about-face for Microsoft; during initial NT development, NetWare was declared "last year's technology" [31].) It will support true pre-emptive multitasking for multiple 16-bit Windows sessions.

It will also have Chicago technology, because some source code, such as that for networking and the UI, is shared. It may also have some technology from Cairo (the object oriented system due out in 1995), perhaps distributed OLE.

7.2. Windows 4.0 -- Chicago

Chicago, while not as new or advanced as NT, marks an important milestone for Microsoft, because it will be its first mainstream low-end PC operating system not based on MS-DOS. It will be Intel-only, and will only run on machines of recent vintage (386 and later), which have the hardware support necessary for a modern operating system. Unlike Windows 3.1 it will support a flat 32-bit address space and multitasking. Unlike NT it will not have security or multiprocessing. It will have a new graphical user interface [89].

Its release schedule is becoming more definite ([9], [18], [88]). In August the first developer's version shipped, PDK (Professional Developers Kit) 1, a 16-bit placeholder. This month PDK 2 started shipping, which has most of the 32-bit plumbing. Next February the first end-user beta is scheduled, with a first pass at the final GUI and with 32-bit OLE. In May or June the second beta should ship. Commercial release is now scheduled for October or November (having slipped from September).

PC Week examined an early beta version ([90], [91]). They noted that "the initial beta release appears more stable and more complete than anticipated." It indeed runs in 4 megabytes. The new GUI borrows from HP's NewWave, Next's NextStep 3.0, IBM's Workplace Shell, Apple's System 7, and OSF's Motif. There is an ever-present "tray" window similar to Apple's top bar. The current Windows 3.1 Program Manager and File Manager are merged into a new tool called the Explorer. File links are supported, as are long file names (as well as compatibility with short DOS file names). System configuration is done through a GUI similar to NT, but the old MS-DOS files (autoexec.bat, etc.) are retained for compatibility. Networking, in the form of a NetWare client clone, is included, as is currently the case with Windows for Workgroups. Plug and play for hardware adapters and drivers is supported, as are universal modem drivers (which are similar to the printer drivers in Windows 3.1).

The new GUI requires a new API, Win32c, the Chicago version of Win32 [91]. It defines about 30 new functions and 60 new messages for the new interfaces. These new functions and messages will be added to NT (and presumably the definition of Win32). Win32 APIs unimplemented in Win32c will have error stubs or null behavior. NT X86 executables will run on Chicago.

For file and print servers, Microsoft is considering a server option for Chicago, along with a reduced version of NT. Sources say beefing up Chicago is favored.

Based on the comments of PC users about NT, Chicago seems to be what many of them want. The analysts and reviewers also favor Chicago. "Chicago may be the operating system you're waiting for... which will have the preemptive multitasking that is probably NT's greatest attraction" [24]. "If hardware prices continue to plummet and a new class of NT-based personal productivity programs emerges, Windows NT may one day reach most desktop users." But "if Chicago does most of what NT does, then NT will always be a niche product" [27].

At the recent Microsoft Developers Conference more details were released, and they indicate some potential weaknesses [18]. The 4 megabyte memory limit has been a critical parameter, forcing some tradeoffs. All the Chicago APIs are 32-bit, but some run in 16-bit space. Brad Silverberg, vice president of the Personal Systems Group at Microsoft, explained that going to an all 32-bit implementation would require up to 30% more memory. This 16-bit implementation may mean slower performance, no pre-emptive multitasking, and vulnerability to crashes. An unidentified IS manager at a Fortune 500 company said, "If they cut corners on [true pre-emptive multitasking] that ... would raise some serious questions. Our biggest problem with Windows today is that it is not pre-emptive multitasking."

7.3. Windows for Workgroups

Windows for Workgroups is an existing intermediate product between Windows 3.1 and NT [92]. It has built-in networking and some 32-bit extensions for increased performance (of up to 30%). It is "a stepping-stone to Chicago" [92].

In the short term, until Chicago is released, Microsoft is positioning the current version, 3.11, as a mainstream Windows platform for 1994. It has negotiated new agreements with hardware OEMs to preinstall 3.11 on their machines. Some nine vendors, including Gateway, DEC, and Zenith, will ship 3.11 on all machines; others, including Dell, will make it an option. Microsoft estimates selling up to 6,000,000 copies

next year; 1,000,000 were shipped last year [92].

8. Software Development for NT

Although in its early stages, software development for NT is not especially painful. Decent programming environments exist, and porting to NT is not "technically challenging" [19]. Further, Microsoft is moving to simplify the various Windows APIs that are emerging.

8.1. Development Tools

PC Week lists 47 vendors with software development tools available or planned for NT, ranging in price from \$29.95 to \$35,000 [19]. Forty-four are scheduled to be available by the end of 1993, with 30 available in September. These products include libraries (for, among other things, icon management, drag and drop support, and X), editors, compilers (for Ada, C, C++, and Fortran 77), programming environments, GUI development environments, tools for code and project management, and software testing, and Unix tools. All of the products will support the Win32 API, and 33 will support the Win32s subset. Two of the most important products are Visual C++/NT from Microsoft and Borland C++.

Visual C++/NT is the NT version of Microsoft's environment for building Windows applications. It has been praised for its performance, profiling features, help facilities, documentation, and most interesting component, the Class Wizard [25]. When the user wishes to create a subclass in C++, the Class Wizard automatically creates the framework of the subclass; the user then simply adds the necessary subclass procedures. The primary problem with Visual C++/NT is that it only supports the Win32 for Intel Platforms. It does not support Win16 or MS-DOS applications, or non-Intel NT platforms.

Borland's C++ tools are now available for NT (the product was scheduled to ship in mid-November) [93]. The product includes an integrated development environment (IDE) with a graphical debugger, an upgraded object oriented window library (OWL, competitive with Microsoft's Foundation Class library), a point and click application generator tool, and a new code generator. Both Win32 and Win32s APIs are supported. It also includes a 16-bit compiler for DOS and Windows 3.1 applications.

8.2. Porting to NT from Windows 3.1

Native 32-bit software products for the Win32 API have been slower to appear than some expected (see the section on NT software products). One explanation for why quick, inexpensive Win32 ports haven't appeared is because users want more in such ports, including multithreading, support for long file names, and MIPS and Alpha versions [2].

Another explanation is proffered by Robert Lee, a manager at Lotus [19]. "Getting up to 32-bit is full of minor pitfalls, but is not technologically challenging." The problem is that the optimizations previously performed to exploit a 16-bit structure are lost. "And almost all applications are pretty seriously optimized." Lewis Levin, the manager of Excel at Microsoft, echoes this, saying that conversion to full 32-bit operation, and dealing with byte alignment in particular, especially for RISC processors, is the first task to undertake (followed by adding threads).

8.3. Porting to NT from Unix

Porting to NT from Unix is made relatively easy for two reasons. First, Unix applications are 32bit, and thus avoid the 16-bit conversion problems of Windows 3.1 applications. Second, "just about everything in the standard C libraries of Unix is available in the C Run-Time Library of NT" [4].

Nonetheless, there are system calls and features that either are not supported in NT or are supported with a different mechanism, and an article in *SunWorld* details some of them [4]. Processes, terminal I/O, security, and file directory manipulation, for example, are substantially different. And subtle differences exist in common system calls (such as file open).

Unix ISVs, however, are used to dealing with such differences and obscurities. Any port to a different version of Unix involves investigating and solving exactly these kinds of problems. The Unix vendors are aware of this and are trying to address the issue with various standards -- see the section on Unix. The differences between Unix and NT do not keep the author of the *SunWorld* article from concluding that "NT is a Windows that even Unix folks could love" [4].

A porting tool exists that helps Unix ISVs deal with some of these problems. Called Nutcracker, from DataFocus, it translates POSIX and most (90%) SVR4 system calls to NT (the percentages are 85% for HP/UX and AIX) [94]. Informix used the tool in their port, and they estimate it cut the time required in half. The tool was to be generally available at the end of 1993 (for the Unix market price of \$9500).

Unite, from Consensys, is another tool providing similar functionality [95]. It consists of libraries supporting Unix APIs, commands, utilities, and development tools, and works on the X86, MIPS, and Alpha architectures. It is priced from \$400 to \$700.

In addition, support exists to ease the transition to the Windows GUI. This support takes the form of X libraries (such as Congruent's NTNIX) that work for Windows, and portable GUI toolkits that support Windows NT (such as XVT) [4]. (Robert Lee of Lotus observes that, for applications he has examined, 1/3 of the code is devoted to the GUI API, a considerable amount to port.)

8.4. The Windows APIs

There are currently three Windows APIs, Win16 (for Windows 3.1), Win32s (a 32-bit enhanced version of Win16), and Win32 (for Windows NT). Looming on the horizon are a new API for Cairo, the next generation of NT, and Win32c for Chicago, with its 30 new functions and 60 new messages.

In order to simplify the API situation and make planning easier for ISVs, Microsoft has announced plans for the development of the APIs [96].

The focus will be on the Win32 API. It will be the canonical API that will include all functionality. It will be expanded to include Win32c when it appears, and will be expanded further when Cairo is released. Developers are being encouraged to write for Win32 through the implementation of all Win32 functions, at least trivially, in all APIs. Every Win32 function will, at the least, return an error.

Feature	Win32s	Win32c	Win32
Win16 (stretched to 32)	x	x	x
32-bit memory management	x	x	x
File mapping	x	x	x
Networking (NetBIOS, Sockets)	x	x	x
OLE 2.0	x	x	x
Mailslot, Named Pipes	-	x	x
Win32 threading	-	x	x
Advanced GDI (Beziers, paths)	-	x	x
RPC	-	x	x
GDI transforms	-	-	x
Event logging	-	-	x
Security	•	-	x
Unicode	-	-	x

For clarity, Microsoft has also provided the following feature chart:

9. OS/2

OS/2 is to a large extent what IBM says it is: Chicago today [10]. It provides a well-regarded GUI (the Workplace Shell, considered better than Windows 3.1), robust 32-bit memory management, and Novell NetWare support [39]. It is also a mature platform attractive to corporate managers [97]. It does, however, require more resources than Windows 3.1, and potentially more than Chicago.

It has also been selling well recently, with monthly sales estimated at between 200,000 and 250,000. It is fourth in *PC Computing*'s Software Saleswire [33] (see the table in the section above on NT sales), although it follows QEMM-386 7.0, a memory management utility. Its sales may surpass Mac OS sales in 1993 [98]. IBM estimates its installed base at more than 4,000,000 at the end of November [10], with 2,000,000 of those the most recent version (2.1), and with 75% of sales to new buyers [11]. Jim Manzi of Lotus says, "We're not about to miss the OS/2 craze, which we think is about to begin" [10]. The craze, by the way, is "a phenomenon that observers attribute in part to the initial negative feedback on Windows NT" [6]. And Brinton Baker, a manager at Oracle, has "a lot of confidence in OS/2", and says that it is "competitive to anything else out there, including NT" [99] (Oracle is releasing a low-end data base product for OS/2). Computer Associates has ported its SuperProject project management software to OS/2, and is shipping one set of disks for both Windows and OS/2.

On the other hand, various analysts are sceptical. Fred Dunn, vice president of a market research firm, says, "OS/2 is used by Fortune 1000 companies for in-house development. For [the mass market] it's not viable at this point" [22]. Brent Williams of IDC says, "doubling the run rate of any operating system in the current market is a very aggressive forecast" [100]. Alex Resnick, president of a consulting firm, believes that "in the end it's just a marketing game because in a year Chicago will be out" [100]. WordPerfect recently halted work on the 32-bit OS/2 version of WordPerfect, a business decision based on the size of the OS/2 market [101], although the company has described OS/2 as a "viable product" [102]. Micrografx recently decided that it can't afford to write for OS/2, even though it had two OS/2 products [11]. And Microsoft, based on royalties paid, estimates only 2,500,000 copies have been shipped, most preloaded on IBM PCs, with no more than 1,000,000 copies in use [11].

IBM appears to be putting its weight behind OS/2. Louis Gerstner recently said that "nothing is more critical to the IBM company than OS/2 and AIX" [6]. Correspondingly, the longer-term IBM operating system efforts, Taligent and Workplace OS, have been subordinated [98]. The Workplace OS, similar in concept technically to NT, "is now a long-term complement to, rather than a replacement for, OS/2 and AIX" [6].

IBM has extensive plans for OS/2 ([10], [103]). A version that will fit in 4 megabytes of memory was to be available in November of 1993. At the end of the first quarter of 1994 a Service Pack upgrade should be available, fixing about 500 bugs. SMP support is also scheduled to appear in the first quarter of 1994. A new 2.2 release is planned for the third quarter of 1994. OpenDoc and Win32s support, Taligent technology, and plug and play are to appear in 1994. OS/2 will also appear on the PowerPC [98].

Jim Louderback of *PC Week* thinks the SMP version in particular is significant, since in OS/2, unlike NT, each Windows task is a separate process and assignable to an individual processor, thus bringing scalability and crash protection to the desktop [104]. "Shipment of OS/2 2.1 SMP will signal the start of the real battle with NT. NT makes a decent server platform, but pales on the desktop. The only way IBM could screw this one up is by pricing the SMP version out of the desktop market," he says. "I used to be ambivalent about OS/2's chances in the market, but no more. If IBM can follow through on its SMP promises and delivery schedule, and price it accordingly, it'll have a winner both on the server and on the desktop" [104].

In addition, IBM is trying to attract Windows users with a version of OS/2 for Windows [105]. OS/ 2 for Windows provides users the familiar Windows desktop and runs Windows programs, and can be installed with Windows 3.1. It does, however, have noteworthy drawbacks. Its resource requirements are those of OS/2, which are less than NT's but more than Windows', and there are compatibility problems with some Windows applications. One analyst characterizes the product as "still essentially IBM in a holding pattern" [104].

IBM is also attempting to lure Windows users to OS/2 with OS/2 multimedia products ported to

Windows (which perform videoconferencing, voice recognition, video compression, and multimedia presentations) [22]. IBM hopes that users will like the products and want to run them on a better platform, namely OS/2. The porting is also an acknowledgement, however, of the importance of Windows. "The ability to use IBM multimedia services under Windows is more attractive that being forced to consider OS/2, buyers said" [22]. In IBM's own third party multimedia program, less than 20% of the products run natively under OS/2 (the vast majority, more than 80%, are for DOS or Windows).

One long term problem for OS/2 is support for the Windows APIs. IBM's agreement with Microsoft expired in September; IBM only has rights to Microsoft's code developed through that month [102]. Now IBM will have to engineer compatibility on its own (in a manner similar to the Unix WABI effort described below). IBM will support Win32s [22], but, as one consultant notes, "to try and maintain Windows compatibility in the long run is a strategy fraught with problems" [102], as Sun has discovered ([106], [107]).

10. NT and the Unix Community

Unlike most PC products, Windows NT and its hardware platforms are being taken seriously by Unix vendors and users. A manager at HP says, "We cannot ignore the existence of NT or its gradual evolution, nor can our partners" [45]. In fact, NT appears to be taken more seriously by the Unix community than by the PC community, because it is fundamentally closer in implementation to Unix than to Windows. Its user interface, however, does make it attractive to "a class of Unix users that doesn't want to run Unix" [53].

In general the reaction of Unix users and ISVs to NT has been positive. In a review of two NT platforms (the NeTpower Series 100 Model 200/67 and the DECpc AXP 150) in one Unix-oriented publication the reviewer notes that both of these platforms have the "performance and features of a typical Unix workstation" (Ethernet, SCSI, TCP/IP, and CD-ROM) and that they "blur the old PC-workstation boundaries" [108]. He concludes that each is an "excellent personal workstation." Another Unix publication concludes that "... the ability for these two operating system environments to coexist in an enterprise-wide environment appears promising" [109]. In an article in a Sun-oriented publication about porting from Unix to NT the author concludes that "NT is a real operating system.... NT is a Windows that even Unix folks could love" [4]. And in a survey of 780 IS managers conducted by X/Open, NT was considered about as "open" (with a rating of 3.6, on a scale of 1 -- not to 6 -- very) as many versions of Unix (HP/UX rated 3.7, SCO 3.7, and Solaris 3.7; AIX rated 2.4, UnixWare 3.3, and OSF/1 4.3) [12].

NT is relatively immature, however, compared to Unix. One Unix user observes that NT is "slow, cumbersome, and deficient on features" [40]. Unix currently is a richer software development platform and a true multiuser platform, has complete TCP/IP networking services, and has cheap file, print, and mail server capabilities out of the box [110]. Phil Hester, a Vice President of IBM, says, "Unix is a hell of a lot more mature than any brand-new operating system, regardless of who's building it. So the high-end systems, in my opinion, are going to be Unix-based for a long time" [71]. It remains to be seen how long it will take NT to mature.

On the other hand, new X86 versions of Unix have been less successful than NT. Only 15,000 copies of Solaris and 30,000 copies of UnixWare have been shipped [40]. They "have not won favor with many corporate users"; they, "like other desktop Unix products, are being relegated to niche and specific vertical applications" [40]. NT may have had an effect; David Smith of IDC observes, "The likely impact of NT is not that it will displace Unix, but rather that it will slam the door on Unix's access to the PC upgrade market" [13]. Michael Slater notest that "It's too late for Unix to become a high-volume product. The chances for Unix to ship millions of units a year have ended" [51].

Indeed, Unix has three fundamental problems compared to the competition (see the comparison survey in [23]): complexity of use and administration, lack of uniformity among implementations (see standards, below), and price/cost [51]. A Windows emulator for Unix (SoftWindows, described below), for example, is priced higher than NT [111], and FrameMaker, which runs on both Unix and Windows, charges 67% more for the Unix version (\$1500 versus \$900 [112]). A vice president at Sun estimates that it costs \$100,000,000 per year to maintain (keep competitive) an operating system, and \$75,000,000 to maintain a microprocessor architecture [13]. Given the volumes of the Unix vendors, those costs are nontrivial on a per unit basis.

The Unix vendors are, nonetheless, so far holding their own; Sun's installed base just passed 1,000,000, shipping 48,000 more systems in 1993 (256,000) than in 1992 [13]. To put this in perspective, Sun is the market leader, with an (IDC) estimated 38% of the workstation-server market (compared to 18% for HP, 12% for DEC, 10% for IBM, and 6% for SGI). For a bit more perspective, Sun's market is 35% commercial, 30% technical, 23% software development, 7% education, and 5% industrial [13]. It will be instructive to track these numbers as NT matures.

10.1. Unix: The Standard Response

Unix has responded to NT by circling the wagons and trying once again to come up with standards, in an attempt to present Unix as a single alternative to NT. Past Unix standardization attempts (such as ACE, Unix International, and OSF) have been unsuccessful. As Mike Azzara, of *Open Systems Today*, notes, "[Unix] vendors assumed they simply could update their old account-control model of the computer business if they could only gain control over enough standards.... [They] determined to slug it out in the market for the right to be the one to set [the] standard interface with their own implementation" [113]. The situation is different now with a common enemy.

First with the Common Open Software Environment (COSE) earlier this year, and then with the Common Operating System Applications Programming Interface (Common OS API), the Unix vendors are uniting. The X/Open organization is being empowered to [114]:

- track and review COSE standards, including the Common Desktop Environment based on Motif;
- brand Unix, based on the Common OS API standards;
- brand Unix System V (from Novell); and
- brand and license Motif (from OSF).

("Branding" here means certifying products compliant, which in turn enables vendors with certified products to use the relevant trade names.)

Nonetheless, it is not clear how successful this effort will be, for three reasons. First, Unix vendors are historically very competitive and will standardize as little as they feel they can get away with. Second, Unix users distrust standardization because of past fiascos. Third, even if Unix becomes standardized users may move to NT because of its independent merits.

A separate effort has been mounted to establish a common Windows interface for Unix, free of Microsoft, called the Public Windows Interface (PWI), with an emulator called the Windows Application Binary Interface (WABI) [115]. It would considerably broaden Unix's appeal to be able to run Windows applications as readily as NT. This effort has, however, encountered technical and business problems.

10.2. COSE and the Common Desktop Environment

Motivated "by the specter of NT eating their lunch" [113], COSE was established by HP, IBM, Univel and AT&T's USL (now combined into Novell's USG), SCO, and SunSoft (DEC has now also joined [115]), in an attempt to provide a common Unix platform.

COSE originally had six components [115]. Those components, and their status as of early November, are:

- The Common Desktop Environment (CDE), an effort to establish a single X-based user interface. Alpha code was sent to developers in October (see below).
- Networking, aimed at establishing uniform network protocols. OSF/1's DCE, SunSoft's NFS, and Novell's NetWare are under review. (OSF may put NFS into DCE in return for Sun's support of OSF [116].)

- Object services, involving standards for distributed objects and associated services. COSE is working with the Object Management Group (OMG) on the Common Object Request Broker Architecture (CORBA).
- System Management, including network administration. COSE is working with X/Open, and has issued no progress reports.
- Multimedia. COSE is working with the Interactive Multimedia Association on Distributed Media Services and the Desktop Integrated Media Environment.
- Graphics. COSE is focusing on Xlib (pixel graphics), PEXlib (2D/3D geometry graphics), and XIElib (advanced imaging) standards.

COSE has recently added three new components [115]:

- A federal network naming service. SunSoft's NIS+ and DEC's CDS (common directory services) are being examined.
- Data management. COSE is working with the Data Management Interface Group (DMIG) on a Unix API.
- Windows on Unix. COSE has adopted the PWI effort.

So far, and not surprisingly, progress has been made primarily in non-controversial areas (object services, multimedia, graphics, data management). In contrast, no apparent progress has been made in system management.

The most significant area from the user's perspective, the one with the most vendor commitment, and the one farthest along, is the CDE. It is also a response to Unix's most obvious weakness compared to Windows -- its varied and weak graphical user interfaces.

The Common Desktop Environment is "pretty much HP VUE in an X11R5 and Motif 1.2 environment" [117]. It consists of six specific elements [118]:

- an enhanced version of the Motif window manager;
- the Hewlett-Packard VUE desktop;
- a help system;
- inter-application data exchange (including cut and paste, drag and drop, and SunSoft's ToolTalk);
- printer management; and
- tools for developers and users, such as a mailer, a calendar manager, a clock, an X terminal and a shell (a windowing version of the Korn shell).

A longer term goal of the CDE is to support editing compound documents, similar to the functionality OLE 2.0 provides.

An application may be integrated into the CDE at one of three levels [117].

- At the lowest level, its launch is integrated, so that it can be invoked via point and click.
- At the next level, it is style-guide compliant, looking and responding like standard applications.
- At the highest level, it is integrated with the desktop services, including the session manager, the workspace manager, the help system, the drag and drop facility, the print services, the style manager, the CDE custom widgets, and the inter-application messaging facility (ToolTalk).

An early developer's version of the CDE, the 10/29 Snapshot, was examined by *Open Systems Today* [117]. The 10/29 Snapshot is "80% stable and complete." "Will CDE cure what ails the Unix industry? Not likely." Nonetheless, "perhaps it's the threat of Microsoft Windows on every desktop, or maybe it's just family togetherness, but it's clear from the 10/29 Snapshot that SCO, Novell, IBM, Sun and HP are

determined to deliver a robust, working desktop environment that draws from the best of the X-Window technology offered by the participating companies."

A crack in the COSE-CDE facade has recently appeared, however. Sun has acquired an interest in Next, is having NextStep ported to Solaris, and has made the commitment to make NextStep, "over time", the default GUI for Solaris [119]. Playing the standards game, Sun is having Next prepare a spec based on NextStep 3.2, called OpenSpec, which will be submitted to X/Open next year [120].

This perceived move to NextStep, away from COSE, has led to renewed scepticism about Unix standards. Mike Azzara notes, "It seems the vendor rule is you can only push hard on a public standard if you have some differentiating factor off in the wings that can make you special again" [121]. "Not coincidentally, CDE is loaded with HP technology," Rob Tholemeier, an analyst with Meta Group, notes. "We think [Sun's decision to support NextStep] will make people hold off going to CDE" [119].

The scepticism has led to clarification by Sun. Ed Zander, of SunSoft, said, "We're fully committed to CDE, and for many years, it will be our dominant platform. But we see [NextStep] technology coming in over the next decade" [122]. Scott McNealy asserted that NextStep will be CDE compatible, and made the "Zen-like observation" [123], "People say 'How does this relate to COSE?' It is. It is. COSE is not an organization, it's a process. It's a verb, not a noun." Nonetheless, there are a "host of incompatibilities" between the Sun and Next platforms, and no availability date for NextStep has been announced [123].

10.3. The Common OS API

An obvious area not covered by COSE is the general operating system interface as defined by "standard" C libraries. To rectify this problem, Sun, HP, IBM, and Novell, among others, formed a group to define a single Unix API [124]. This API is being designed so that 90% of all applications will port to another Unix platform simply by recompiling [125]. The idea is to replace the traditional Unix source code license with a branding process [126].

The API is based on three sources [124]:

- the X/Open Portability Guide, Edition 4 (XPG4),
- the System V Interface Definition Version 3 (SVID3, from USG), and
- the Application Environment Specification (AES, from OSF for OSF/1, which extends POSIX).

Using those starting points, the system interfaces used by "bellwether applications" from 10 vendors (Autodesk, Cadence, Frame, Lotus, Informix, Island Graphics, the SAS Institute, Sybase, Teamworks, and Word Perfect) were examined, and an initial API was defined [125]. Applications from 50 additional vendors were then examined, and more interfaces (15% more) were added. The resulting standards are for 1170 kernel items: 960 system interfaces, 174 Unix commands, and 70 header files [127]. These standards are now collectively known as "Spec 1170". The "Common OS API" refers to the process (which may be ongoing) as well as the standard.

A number of significant areas are not covered: real time, graphics, threads, security (beyond chown), extended utilities (awk, sed, shells, printing, and tools for system administration), word sizes, and byte ordering [128].

A draft specification was completed in September; Unix International and OSF were to complete a review in November of 1993; a final specification from X/Open is due in November of 1994 [125]. OSF will build the test suites. "Most" major vendors (including IBM, SunSoft, USG, and SCO) could comply with the proposed specification by mid-1994 [127]. The group is trying to put some weight behind the standard; the approval of 75 vendors and developers is sought [127].

Novell has officially transferred the Unix trademark (valued at \$15,000,000 [129]) to X/Open for branding purposes [20], without any strings [130] (initially, in contrast, Novell had proposed branding based on the UnixWare source [131]). Novell has retained rights to the System V source [20]. Until the standard is finalized, interim branding will be done using XPG3 and XPG4 [126].

10.4. The Response to the Common OS API

It appears that the Unix vendors are unified as they have not been in the past. Nonetheless, based on that past, scepticism is rampant among analysts and users. Headlines include "The Common OS API Is Nothing More Than Fear Of Microsoft's Windows NT" [128], and "Unix Vendors Sing the Same Tired Tune (Again)" [21]. It's "just a lot of loud public relations" [128] from the "fractious Unix community" [131]. It's "100 percent public relations" (Scott Winkler of the Gartner Group) [20]. One MIS manager complains, "Everyone in Unix has a big ego, and no one will agree to a standard" [131]. Another says the standard is "not in the same league as NT" [132]. One commentator calls it "a gruesome behemoth" [110]. Another points out that the list of areas not covered (presented above) is substantial and significant to software vendors. Yet another doesn't believe in standards [12]: "It doesn't matter whether you call it open, proprietary, international, or de facto. There's no such thing as a real standard. What matters is market share."

Banyan, the network ISV, on the other hand, is positive. It produces Vines for SCO, Solaris, AIX, and HP-UX, using 160 programmers. Currently thirty of those programmers simply maintain uniformity and compatibility, positions that would be unnecessary with Spec 1170.

Rikki Kirzner of Dataquest believes the standard will have an effect, but a limited one [21]. "It could increase sales on the server side of Unix and could slow NT's penetration down by a year, but it's not going to stop it."

10.5. The WABI Initiative

A serious limitation of Unix, in contrast to NT, is its inability to run Windows and DOS applications. For some time various products have addressed this issue, the most notable being the SoftPC emulator from Insignia, but they have not been part of Unix.

Spearheaded by Sun, an effort, called the Public Windows Interface (PWI), is being undertaken to give Unix this functionality. Its goals are:

- to provide complete API compatibility with Windows;
- to do so without emulating the X86, for higher performance;
- to do this independently of Microsoft; and
- to do so with a package shipped free with each copy of Unix.

The specific Windows emulator being developed is the Windows Application Binary Interface (WABI).

WABI is being developed jointly by the SunSelect subsidiary of Sun, by the USG component of Novell, and by IBM [133]. A number of Unix vendors and Windows ISVs are part of the PWI group, including Sun, Borland, WordPerfect, Corel, IBM, Novell (through USG), HP, SCO, and Quarterdeck [106].

WABI version 1.0 started shipping in November with Solaris 2.3 [134]. Sun has had trouble moving users from SunOS to Solaris 2.x, and WABI could be a "deciding factor" in motivating the move [55]. HP will ship it within a quarter, and it is scheduled to appear with AIX on the RS/6000 and PowerPC early next year [133]. It will also be shipped with Unix versions from USG and SCO [135]. Parenthetically, Sun is shipping WABI for free, but unbundled, requiring that Solaris 2.3 users request it, in order to gauge demand [134].

The current WABI, however, has notable limitations. It only supports the Win16 API [136]. The Win32s and Win32 APIs, as well as DOS applications and networked and multimedia products, won't run ([106], [137]). In addition, the initial version is not very robust. "Nearly all of the 13 certified applications encounter various problems when operating with Wabi, according to the release notes" (in an article titled "SunSoft to ship buggy Wabi") [138]. The certified applications include Excel, Word, PowerPoint, Procomm Plus, Corel Draw, and Aldus Pagemaker [137]. A *PC Week* test of WABI found it "a limited product" and concluded that "its immediate success will depend on how inexpensively end users can obtain it" [137].

Original supporters have become critical or uninterested, including Borland, WordPerfect, Novell, HP, and Lotus [106]. Not surprisingly, Microsoft is taking a hard line against WABI. It is making clear to its "large-account" customers that it does not certify its applications on WABI [139]. Dale Christiansen, a man-

ager at Microsoft: "If a user calls and says, 'My data's corrupted,' there's nothing we can do about that" [107].

WABI also has a serious competitor in SoftWindows, Insignia's Windows-oriented follow-on product to SoftPC [140]. It runs Windows applications 30% to 50% faster than SoftPC [141], and is licensed by Microsoft [133]. It is not limited in the Windows applications it runs, and it supports NetWare, TCP/IP, NFS, and DOS [137]. It is two times faster than WABI on CPU and disk benchmarks, although its screen I/O is three times slower [137]. It is scheduled for release in December on Solaris and HP/UX, and in March on AIX, Irix, and Mac OS on the PowerPC [141].

In addition, there are other Windows-on-Unix tools [142] some licensed by Microsoft ([139], [143]). Some are software development tools for converting Windows applications to Unix (MainSoft's MainWin, Quorum's Equal Windows, and Bristol's Wind/u), one is an emulator for X86 Unix systems (Locus' Merge, offered by Sun on its X86 version of Solaris 2), and one makes Motif applications running on Unix hosts look like Windows applications when accessed through a PC X server (IXI's Win-tif).

On the other hand, IBM appears squarely behind the WABI technology (which makes sense given its differences with Microsoft). It has contributed its PC-SIM MS-DOS simulation technology [143], and wants a 32-bit version [106]. In addition, Novell is helping with the network technology [106]. And the next version, 2.0, will include support for 17 more applications and DOS emulation [143].

Still, "many are skeptical Wabi will achieve sufficient critical mass to keep going" [106]. "The bottom line: Today's Wabi simply doesn't do what it needs to do to be competitive, and most industry players aren't willing to wait until it does" [106].

11. What About Apple?

Apple is what Next was, a hardware company distinguished by its software. Apple, at least in the near term, is successful enough that it will not change itself as Next has. Michael Spindler, the new president of Apple, says that "the one thing we can't risk is bringing Apple to its knees by open licensing the OS and trying to turn to a software-only business" [144].

What this means in terms of Windows and NT is that Apple is focused on defending its hardware/ software niche, much like most Unix vendors. Unlike Unix, however, there is no X86 version of Apple software to compete for PC platforms. Also unlike Unix, Apple does not have high-end SMP server technology.

Currently, Apple is absorbed in keeping its current customers. Crucial to this is the move from the 680x0 to the PowerPC processor, which will improve the price/performance of Apple's hardware [144]. It is also evolving Mac OS, with a new version supporting, among other things, inter-application drag and drop, a scripting language, and portable documents (using a format that is not application specific) [145].

Additionally, like the Unix vendors, it is trying to keep existing customers and attract new ones with support for Windows applications [146]. It is doing this through add-on hardware, with an extra \$500 board, complete with a 486 chip, that will run PC software.

In sum, Apple is fighting to preserve its part of the desktop hardware business. Unlike IBM with OS/2 and the Unix vendors with their X86 versions, it is not trying to spread its software, and is in no rush to license its technology. Michael Spindler [144]: "First we take care of the installed base [with the PowerPC], [then] move beyond and acquire new customers with this Windows Inside thing. Then we say who we are licensing and how."

Mac OS does not appear to challenge Windows, and certainly not NT. In contrast, Chicago, with its improved GUI, is something of a threat to Mac OS, since Apple's primary technical advantage is its superior GUI.

12. Summary: Corporate Strategies

With NT and its Unix-like technology, including TCP/IP, Microsoft has effectively blocked the spread of Unix to the desktop (particularly its X86 variants), and has gotten the attention of Unix vendors, and, more importantly, ISVs.

The hype about NT may have raised unrealistic expectations, but it also brought momentum and a lot of interest in the product. Expectations are now more realistic. That makes NT a relatively low-risk way for ISVs to port to Win32, with 32-bit applications in existence when Chicago arrives.

The portability of NT has drawn in many major computer companies with the promise of a piece of Intel's market; the prospect of hardware revenues has caused the software parts of these companies to commit resources to NT. DEC, Motorola, HP, and Intergraph are doing their own NT ports. DEC is porting IBM network software to NT, and HP is porting its own network software.

One aspect of NT that has given it credibility but has not been much of a factor commercially is its ability to run 16-bit Windows applications. Users do not seem interested in acquiring NT in order to run such applications on a more robust platform. OS/2 can do that with less hardware.

So far Microsoft has been successful at evolving the Windows APIs (which ultimately give it control of the desktop). Because of the specialized optimizations required by 16-bit applications, the general move to 32 bits appears at least as difficult as the move to the Win32 API. Although there are not currently many 32-bit Windows applications, Microsoft produces several important 16-bit applications, and thus guarantees the existence of significant 32-bit apps. Getting such applications is a problem for IBM with OS/2.

Microsoft has also been successful at controlling the APIs. It encourages other application developers to use them, and licenses other implementations of them when necessary (OS/2, for example, and Soft-Windows). Nonetheless, it controls the development of the technology and many of the major products.

Still, Microsoft is to some extent vulnerable now, before Chicago's release and with NT, at this point, an immature niche product. The negative press about NT has drawn positive attention to OS/2.

IBM, with its heightened enthusiasm for OS/2 and its announcement of the PowerPC, is trying to take advantage of this weakness. It has a chance to make inroads against Microsoft, and Intel, especially with its (IBM's) traditional corporate customer base. OS/2 and the PowerPC, along with AIX, give IBM the one-stop shopping solution that, in the past, was its hallmark with large corporate customers.

The problems for OS/2 are that: there are a limited number of native applications; it has a narrow window of opportunity, before Chicago arrives; and Microsoft still owns the Windows APIs (and will evolve them with Chicago). Although IBM sells OS/2 as "Chicago today", most users are likely to wait and compare the two. The problem for the PowerPC is that it is very new.

Another minor factor against OS/2 is that at least some Unix applications will migrate to NT, giving some big system weight to the 32-bit Windows API and NT. Those applications likely will not move to OS/2 because it is less Unix-like and is not seen by anyone as a competitor to Unix.

On the other hand, as long as OS/2 does not become too popular -- as long as it follows the Windows API and most applications are written for Windows -- Microsoft could leave OS/2 alone, letting it serve as competition for antitrust purposes.

Perhaps because of the recent success of OS/2 relative to NT, or for technological reasons, IBM is downplaying its real NT equivalent, Workplace OS, parts of which will ship sometime in 1994. It appears that IBM will take what it can get with OS/2, but that its long term strategy is focused more on the PowerPC hardware than on software. IBM apparently intends the PowerPC to transcend software; supposedly it will, at some point (partly through emulation), run OS/2, Mac OS, NT, Workplace OS, AIX, Taligent, and Solaris.

The big problem for the Unix vendors is that they no longer have technological superiority on the desktop. With OS/2, Chicago, and especially NT and the Workplace OS, many, if not most, of Unix's historical technological advantages no longer exist. Unix, with its small but significant collection of applications, is vulnerable to the migration of those applications.

Unix, however, does have networking technology that in some ways is superior to that of the PC universe. Historically PCs have had no security and no built-in networking, so that PC networks and electronic mail are proprietary add-ons. In contrast, with TCP/IP, FTP, and Internet mail, Unix inherently supports heterogeneous networking. This built-in heterogeneous networking may be an advantage in the construction of new distributed, object based systems.

One party that has not yet been heard from is Intel. NT's portability is an obvious threat. Its immaturity gives Intel some time, but if NT becomes popular and the RISC platforms sell well, Intel will respond. Possible responses include special built-in hardware support for Windows, development of a RISC engine with X86 emulation in hardware, and investment in competitive software (such as OS/2).

A final note on Microsoft: it is attempting to take on a new role with NT (and more notably with Cairo), that of technology leader. It has a credibility problem in this role. It is not known for research or innovation, and it has not done a great job managing the NT vision thing.

13. Summary: Items to Watch for

It is not a simple matter of whether NT will succeed or fail -- it is too new and too visible to do either soon. The extent of its success (and failure) will be affected and indicated by some specific items:

- The number of 32-bit Windows applications. Microsoft is expending substantial effort to get applications ported to NT. It remains to be seen if a critical mass will be reached. Note also that X86 NT applications should also be readily portable to the Chicago API.
- The number of NT desktop applications. NT can expand beyond a server niche; additionally, desktop applications give NT general credibility.
- The number of bellwether Unix applications moved to NT. If a significant number of the bellwether applications used for Spec 1170 move to NT, Unix will be seriously threatened.
- The number of NT applications on non-X86 platforms. This will affect the strength of the NT RISC hardware vendors.
- The number of (X86) machines shipped with NT. This will indicate to what extent NT will become a standard.
- Chicago's final resource requirements. If Chicago is about the same size as Windows 3.1, most Windows users will upgrade. If it isn't, OS/2 will benefit.
- Reviews of beta releases of Chicago. These should indicate anticipated resource requirements, and also give some indication of the quality and features of the final release. If they are very positive or very negative, they may affect OS/2 sales.
- Sales of OS/2. Specifically, note if OS/2 expands out of its old IBM roots in the Fortune 1000.
- The number of OS/2 applications. In comparison with 32-bit Windows applications, this will indicate whether OS/2 is seen as a niche product.
- The condition and acceptance of various intermediate upgrade products. These include Windows for Workgroups, OS/2 2.2, and Windows NT 3.11. The general success of Windows products will adversely affect OS/2 and vice versa.
- Sales of Unix systems. Decreasing sales will probably correspond to rising NT sales. Strong sales of X86 Unix will indicate strength against NT.
- The state of COSE, the Common OS API, and WABI. These standards will determine the appearance and functionality of Unix.
- The appearance of Unix features on NT (either through Microsoft or through third parties). NT is not multiuser, and does not come with a mail system usable from Unix or the Internet. These lacks make it more difficult to integrate into a Unix environment.
- PowerPC Macintosh sales. If such sales are strong, Apple will be able to hold on to its niche.
- The status of object based systems. Cairo, DOE/NextStep, and Taligent may be significant. Whether they will be will depend on specific features and applications.

The general condition to monitor is application migration: which applications, to which platforms.

Users, if they can manage, should wait a year before choosing an operating system or buying hardware. In that time there will be

- Chicago (Windows 4.0),
- Daytona (Windows NT 3.11),
- an OS/2 upgrade (OS/2 2.2),
- Spec 1170 and the CDE,
- increased Pentium sales (it is projected that in a year up to 25% of Intel's chip shipments will be Pentia [14]) and PCI bus hardware, and
- cheap non-X86 NT and OS/2 boxes (Alphas, MIPSes, and PowerPCs).

14. Conclusion

Three systems became significant on desktops the 1980s: Unix, the Macintosh, and MS-DOS. Unix was the most technologically sophisticated, being based on minicomputer and mainframe time sharing. The Macintosh defined ease of use and graphical user interfaces. MS-DOS was cheap, ubiquitous, and did enough to support much of what people wanted to do.

Now, with Windows 3.1 and Windows NT, the Macintosh no longer has such a clearly superior user interface, and Unix is no longer clearly the technological leader. With more powerful PCs and OS/2, NT, and, soon, Chicago, 16-bit MS-DOS will start to fade.

In other words, the technical differences between operating systems is becoming ever smaller, and success will be determined more by market forces and applications than by increasingly indistinguishable base technology.

This environment favors Microsoft, with its installed base of systems and applications. If it releases timely products that are good enough, it will consolidate the advantage it now holds with Windows.

15. Acknowledgements

Thanks to Matt Haley, Ellen Peel, and Jon Pincus for close readings of drafts of this paper, and many good suggestions.

16. References

- [1] "Finally, NT Arrives"; Paul Kapustka; Open Systems Today, 7 June 1993, pp. 1, 74.
- [2] "Just what is Windows NT? Many things to many users"; Larry J. Seltzer; *PC Week*, 20 September 1993, p. 105.
- [3] "DEC, Microsoft relationship relaxes"; Kimberly Patch; *PC Week*, 8 November 1993, p. 159.
- [4] "Porting to NT"; Carl Dichter; *SunWorld*, September 1993, pp. 86-94.
- [5] "With Idea Of Desktop Unix All But Dead, The Right Place To Focus Is On The Server"; Sally Atkins; Open Systems Today, 22 November 1993, p. 65.
- [6] "Cannavino delivers eulogy for 'old' IBM"; Robert L. Scheier; *PC Week*, 22 November 1993, pp. 1, 10.
- [7] "The 100 Most Valuable Products of the Year"; John Montgomery & John Taschek; *PC Computing*, December 1993, pp. 248-252.
- [8] "Windows NT Poses Challenge For Microsoft"; Don Clark; *Wall Street Journal*, 30 November 1993; pp. B1, B7.
- [9] "Microsoft demos Chicago, makes strong push for NT"; Amy Cortese; *PC Week*, 22 November 1993, p. 10.
- [10] "Seasoned OS/2 gains ground"; Amy Cortese; PC Week, 22 November 1993, pp. 1, 12.
- "IBM's OS/2 Is More Successful Than Expected"; Laurie Hayes; Wall Street Journal, 30 November 1993; pp. B1, B7.

- [12] "The State of Open Systems '93"; *Computerworld*, 6 December 1993, pp. 87-122.
- [13] "Open Systems Moves Promise Bright Future", Barry D. Bowen, Open Systems Today, 6 December 1993, pp. SF1, SF10, SF11.
- [14] "Steam Builds On Intel Pentium Rollouts"; Walter Andrews; *Electronic News*, 29 November 1993, p. 4.
- [15] "Microsoft prepares users for extended NT 'gestation'"; Marc Ferranti; *PC Week*, 20 September 1993, pp. 1, 16.
- [16] "When deployed carefully, Windows NT handles the job"; Eamonn Sullivan, Larry J. Seltzer, David Chernicoff; *PC Week*, 20 September 1993, pp. 105, 109.
- [17] "Chicago bulls ahead"; Stuart J. Johnson & Ed Scannell; *Computerworld*, 20 December 1993, pp. 1, 16.
- [18] "Chicago nears 32-bit goal"; Amy Cortese; PC Week, 13 December 1993, pp. 1, 12.
- [19] "From a developer's viewpoint, Windows NT is no Windows"; Avery L. Jenkins; *PC Week*, 20 September 1993, pp. 116, 123.
- [20] "Novell hands off Unix branding to X/Open"; Mary Jo Foley; PC Week, 18 October 1993, p. 20.
- [21] "Unix Vendors Sing the Same Tired Tune (Again)"; UnixWorld, November 1993, p. 15.
- [22] "IBM gives Windows users a splash of multimedia glitz"; Erica Schroeder; *PC Week*, 15 November 1993, pp. 1, 16.
- [23] "NT starts slow as users seek more meat"; Ed Scannell; Computerworld, 15 November 1993, pp. 1, 16.
- [24] "Windows NT: Changing the PC Landscape"; Nicholas Baran; PC World, October 1993, pp. 156-160.
- [25] "Visual C++/NT: Strong but imcomplete"; Peter Coffee; PC Week, 20 September 1993, pp. 125, 129.
- [26] "Is There a Better Windows 3.1 Than Windows 3.1?"; Jon Udell; BYTE, November 1993, pp. 85-96.
- [27] "NT Applications: The Waiting Game"; Eric Auchard; PC World, October 1993, pp. 185-196.
- [28] "SQL Server for Windows NT blazes new trail with performance, tools"; Bob Gallagher; *PC Week*, 13 September 1993, pp. 1, 14.
- [29] "Windows NT goes to production line"; Robert L. Scheier; PC Week, 15 November 1993, p. 20.
- [30] "Microsoft ships bug fix for NT"; Amy Cortese; PC Week, 18 October 1993, p. 16.
- [31] "NT: Is It Ready For Critical Apps?"; J. William Semich, Datamation, 15 May 1993, pp. 28-33.
- [32] "Micrsoft continues all-out NT push"; Amy Cortese; PC Week, 15 November 1993, p. 11.
- [33] "Software Saleswire"; *PC Computing*, December 1993, p. 168.
- [34] "Software Saleswire"; *PC Computing*, January 1994, p. 128.
- [35] "Software Saleswire"; PC Computing, October 1993, p. 112.
- [36] "Software Saleswire"; *PC Computing*, November 1993, p. 154.
- [37] "PC Week Labs salutes the best 1993 had to offer"; PC Week, 27 December 1993, pp. 61-67.
- [38] "Reports of Windows NT's death are premature"; Peter Coffee; PC Week, 1 November 1993, p. 65.
- [39] "It's OK to date NT, but too early to move in"; Christine Comaford; PC Week, 30 August 1993, p. 46.
- [40] "X86 Unix struggles for user loyalty"; Mary J. Foley; PC Week, 29 November 1993, pp. 57, 60.

- [41] "Microsoft outlines memory, speed upgrades for latest NT"; Ed Scannell; Computerworld, 13 December 1993, p. 2.
- [42] "Hey, Bill, Time To Weed Out The Guts From The Grace"; Sam Whitmore; *PC Week*, 15 November 1993, p. A/26.
- [43] "NT garners interest"; Alyson Preston; *PC Week*, 20 September 1993, p. 105.
- [44] "Banyan puts ENS development plans for NT on back burner"; Eric Smalley; *PC Week*, 29 November 1993, p. 6.
- [45] "HP edges toward NT support"; Mark Halper & Jean S. Bozman; *Computerworld*, 13 December 1993, pp. 1, 12.
- [46] "Users and ISVs find rules are changing in client/server game"; Kimberly Patch; *PC Week*, 4 October 1993, pp. 1, 11.
- [47] "Support is No. 1"; Amy Schurr; *PC Week*, 15 November 1993, p. 126.
- [48] "DEC predicts full-speed ahead in 1995"; Kimberly Patch; PC Week, 20 September 1993, p. 176.
- [49] "Microsoft, DEC forge agreement for integrated OLE development"; Amy Cortese; PC Week, 6 December 1993, p. 1, 11.
- [50] "Computer Giants Dual Over Operating Systems"; Wall Street Journal, 30 November 1993; p. B1.
- [51] "The New CPUs: An Interview With Microprocessor Report Publisher Michael Slater"; Mary Eisenhart; *MicroTimes*, 15 November 1993, pp. 73-84.
- [52] "The Labs picks the Not So Nice of '93"; PC Week, 27 December 1993, p. 67.
- [53] "NT Plus RISC Equals A New Breed Of Desktops"; Paul Krill; Open Systems Today, 22 November 1993, p. 44.
- [54] "Comdex '93: System software"; Ed Scannell; *Computerworld*, 15 November 1993, p. 29.
- [55] "SunSoft to bundle Wabi into Solaris 2.3"; Mary Jo Foley; *PC Week*, 20 September 1993, pp. 73, 76.
- [56] "NT ramp-up sluggish despite Microsoft's rosy assertions"; Amy Cortese; PC Week, 1 November 1993, p. 8.
- [57] "NT systems pick up NetWare links"; Amy Cortese; PC Week, 20 September 1993, p. 78.
- [58] "Windows NT still a LAN outsider"; Eric Smalley; PC Week, 20 December 1993, p. 19.
- [59] "NT NFS Product From Beame & Whiteside"; Open Systems Today, 22 November 1993, p. 4.
- [60] "Microsoft unveils SNA NT version"; Jai Singh; PC Week, 8 November 1993, p. 3.
- [61] "Trio bridging NetView, NT"; Paula Musich & Susan E. Fisher; *PC Week*, 15 November 1993, pp. 1, 14.
- [62] "NT captures support of HP, DEC, IBM"; Paula Musich & Stan Gibson; *PC Week*, 22 November 1993, p. 16.
- [63] "SQL Server NT push under way"; Karen D. Moser; *PC Week*, 20 December 1993, pp. 49, 55.
- [64] "OS/2 and Windows Networks"; Barry Nance; BYTE, November 1993, pp. 117-122.
- [65] "Watcom widens server reach"; Karen D. Moser; PC Week, 18 October 1993, p. 73.
- [66] "Informix-SE to appear on NT Server"; Mary Jo Foley; PC Week, 15 November 1993, p. 84.
- [67] "Microsoft ships FORTRAN compiler for NT"; Norvin Leach; PC Week, 6 December 1993, p. 70.
- [68] "32-bit library highlights Symantec C++ 6.1"; Norvin Leach; PC Week, 20 December 1993, p. 6.
- [69] "NT Resource Kit comes to rescue"; Eamonn Sullivan; *PC Week*, 13 December 1993, pp. 89, 95.
- [70] "NCR makes a bid for NT users with OS/2 migration program"; Stan Gibson; *PC Week*, 4 October 1993, p. 6.

- [71] "Launching The PowerPC"; Robert Hertzberg; Open Systems Today, 13 September 1993, pp. 42, 44.
- [72] "NEC, DEC increase NT RISC push"; Susan E. Fisher; PC Week, 15 November 1993, pp. 1, 11.
- [73] "DEC Chip Fuels Fight With Intel Over NT"; Pat Krill; Open Systems Today, 20 September 1993, p. 4.

ng L

- [74] "DEC pushes low-cost Alpha AXP chips for building NT systems"; Susan E. Fisher; *PC Week*, 20 September 1993, pp. 33, 38.
- [75] "DeskStation NT RISC Boxes Get New Chipset"; Ann Miller; Open Systems Today, 22 November 1993, p. 44.
- [76] "First R4600 RISC PCs on tap"; Susan E. Fisher; PC Week, 29 November 1993, p. 20.
- [77] "NetPower rolls out MIPS-powered NT server line"; Susan E. Fisher; *PC Week*, 4 October 1993, p.
 69.
- [78] "Comdex"; *Electronic News*, 15 November 1993, p. 29.
- [79] "IBM jump-starts PowerPC strategy"; Neal Boudette; PC Week, 15 November 1993, p. 8.
- [80] "Intergraph Deal With Sun For NT SPARC Port"; Paul Krill & Mitch Wagner, Open Systems Today, 19 July 1993, p. 43.
- [81] "SMCC, Intergraph join forces to do Microsoft NT port"; Peggy King; *SunWorld*, September 1993, pp. 26-27.
- [82] "New HP sub-\$4000 RISC workstation will run Windows NT"; Susan E. Fisher & Mary Jo Foley; *PC Week*, 6 December 1993, pp. 1, 16.
- [83] "MP PCs target super-server demand"; Stephanie Wilkinson; *PC Week*, 6 December 1993, pp. 115, 119-122.
- [84] "NCR Cuts Prices, Adds Controller"; Scott Friedman; Open Systems Today, 30 August 1993, p. 17.
- [85] "Cisco, Microsoft Cut Router Deal"; Open Systems Today, 22 November 1993, p. 25.
- [86] "Pentium-based PC price plunge continues"; Neal Boudette; PC Week, 15 November 1993, p. 6.
- [87] "Microsoft giving some polish to Windows NT"; Amy Cortese; *PC Week*, 22 November 1993, pp. 41, 49.
- [88] "Microsoft conducts Windows tour"; Amy Cortese; PC Week, 20 December 1993, p. 12.
- [89] "Chicago nears beta testing"; Amy Cortese; PC Week, 30 August 1993, pp. 1, 16.
- [90] "Chicago beta radically new"; Eamonn Sullivan & Larry J. Seltzer; *PC Week*, 13 September 1993, p. 12.
- [91] "Microsoft's Chicago bears watching"; Eamonn Sullivan & Larry J. Seltzer; *PC Week*, 20 September 1993, pp. 20, 24.
- [92] "WFW positioned as main Windows platform"; Amy Cortese; PC Week, 6 December 1993, p. 65, 67.
- [93] "Borland C++ compilers on tap"; Norvin Leach; PC Week, 4 October 1993, p. 6.
- [94] "Nutcracker shifts Unix apps to Windows NT"; Norvin Leach; PC Week, 18 October 1993, p. 71.
- [95] "Consensys unveils Unix-to-NT porting tool with SVR4 code"; Amy Cortese; *PC Week*, 20 December 1993, p. 19.
- [96] "Microsoft aims to simplify Win APIs"; Amy Cortese; PC Week, 4 October 1993, pp. 57, 65.
- [97] "PC Managers Speak Out About NT"; *Datamation*, 1 October 1993, pp. 68-74.
- [98] "IBM reins in Workplace OS as OS/2, AIX sales accelerate"; Amy Cortese & Mary Jo Foley; *PC Week*, 8 November 1993, pp. 1, 14.

- [99] "Oracle announces port of RDBMS to IBM's new OS/2 SMP system"; Karen D. Moser; *PC Week*, 29 November 1993, p. 68.
- [100] "IBM hopes OS/2 for Windows will broaden system's appeal"; Marc Ferranti; *PC Week*, 22 November 1993, pp. 41, 49.
- [101] "WordPerfect halts 32-bit OS/2 version"; Ted Smalley Bowen; PC Week, 15 November 1993, p. 16.
- [102] "IBM faces hurdles with OS strategy"; Amy Cortese; *PC Week*, 13 September 1993, pp. 57, 68.
- [103] "IBM greets 1994 with OS/2 2.1 maintenance fix"; Amy Cortese; PC Week, 20 December 1993, p. 12.
- [104] "The SMP version of OS/2 2.1 could break into the big time"; Jim Louderback; *PC Week*, 29 November 1993, p. 142.
- [105] "IBM's OS/2 for Windows carries attractive price but has its quirks"; Larry J. Seltzer; *PC Week*, 15 November 1993, p. 16.
- [106] "A Wobbly Wabi"; Sam Whitmore; PC Week, 20 September 1993, p. A/6.
- [107] "Microsoft rules out support for Windows apps on Wabi"; Mary Jo Foley; *PC Week*, 22 November 1993, p. 6.
- [108] "Two RISC PCs Set Speed Standards"; Jason Levitt; Open Systems Today, 11 October 1993, pp. 70, 77.
- [109] "Can Windows NT Coexist With UNIX?"; Tony L. Hampel & Anna K. Weldon; Unix Review, November 1993, pp. 43-48.
- [110] "What's Right About NT Makes One Ponder 'What Ifs' For Other Platforms"; Jason Levitt; Open Systems Today, 22 November 1993, p. 59.
- [111] "Insignia to debut Wabi competitor"; Beth Freedman; PC Week, 13 September 1993, p. 3.
- [112] "FrameMaker gets a boost"; Ted Smalley Bowen; PC Week, 13 September 1993, pp. 35, 45.
- [113] "My Own Personal Peloponnesian War"; Mike Azzara; Open Systems Today, 20 September 1993, pp. 30-31.
- [114] "X/Open faces burden of its expanded role"; Mary Jo Foley; PC Week, 11 October 1993, pp. 138-9.
- [115] "COSE consortium to examine additional standards for Unix"; Mary Jo Foley; PC Week, 8 November 1993, p. 69.
- [116] "OSF, Sun bargain over standards"; Mary Jo Foley; PC Week, 20 December 1993, p. 6.
- [117] "An Initial Attempt To Unify Unix GUIs"; Jason Levitt; Open Systems Today, 22 November 1993, pp. 53, 58.
- [118] "Tomorrow's Unix Desktop Defined?"; Mike Carl; Open Systems Today, 5 July 1993, p. 63.
- [119] "Sun adds NextStep to stable of Solaris GUIs"; Mary Jo Foley; *PC Week*, 29 November 1993, pp. 1, 10.
- [120] "Sun Micro Takes Interest in NeXT"; Reinhardt Krause; *Electronic News*, 29 November 93, pp. 1, 16.
- [121] "Anticipating The Next Bombshell", Mike Azzara, Open Systems Today, 6 December 1993, p. 40.
- [122] "Developers await Sun-NeXT details"; Mary Jo Foley; PC Week, 6 December 1993, p. 65, 67.
- [123] "Stiff Test for Sun, Next In Wake of \$10M Deal", Mitch Wagner, Open Systems Today, 6 December 1993, pp. 1, 105.
- [124] "Unix Source Compatibility At Last?"; Mitch Wagner; Open Systems Today, 30 August 1993, p. 1.
- [125] "No Technical Hurdles Seen By Backers Of Common OS API"; Mitch Wagner; Open Systems Today, 13 September 1993, pp. 5, 63.

- [126] "Scepticism Palpable As Trademark Deal Nears"; Mitch Wagner; Open Systems Today, 11 October 1993, pp. 1, 8.
- [127] "Vendors sketch common Unix spec"; John Pallatto; PC Week, 6 September 1993.
- [128] "The Common OS API Is Nothing More Than Fear Of Microsoft's Windows NT"; Andy Feibus; Open Systems Today, 11 October 1993, p. 25.
- [129] "Novell To Relinquish Unix Trademark To Industry"; Mitch Wagner; Open Systems Today, 27 September 1993, pp. 1, 80.
- [130] "Novell to Hand Off UNIX Rights to X/Open"; William Brandel; LAN Times, 18 October 1993, pp. 1, 96.
- [131] "Fractious Unix community snubs Novell's X/Open Plan"; Mary Jo Foley; PC Week, 27 September 1993, pp. 1, 14.
- [132] "Reaction To Unix Unity Move Is Mixed"; Mitch Wagner; Open Systems Today, 13 September 1993, p. 5.
- [133] "IBM Adds Weight to WABI Initiative"; Mitch Wagner; Open Systems Today, 30 August 1993, p. 3.
- [134] "Upgrade News For Reluctant Sun Users"; Mitch Wagner; Open Systems Today, 20 September 1993, pp. 1, 98.
- [135] "IBM backs Wabi software"; Mary Jo Foley; PC Week, 6 September 1993, p. 55.
- [136] "Win updates, Wabi to square off"; PC Week, 6 September 1993, p. 63.
- [137] "WABI: More Limited Than The Competition", Jason Levitt, Open Systems Today, 6 December 1993, pp. 1, 82, 84.
- [138] "SunSoft to ship buggy Wabi"; Mary Jo Foley; PC Week, 1 November 1993, p. 8.
- [139] "SunSelect's WABI Absorbs New Blows"; Mitch Wagner; Open Systems Today, 22 November 1993, p. 3.
- [140] "Insignia Eyes New Height For Its Emulator"; Mitch Wagner; Open Systems Today, 20 September 1993, p. 5.
- [141] "Insignia rolls out Phase I of Windows-on-Unix scheme"; Mary Jo Foley; PC Week, 20 September 1993, p. 75.
- [142] "Win-on-Unix tools debut"; Mary Jo Foley; PC Week, 18 October 1993, pp. 63, 67.
- [143] "Windows-on-Unix entrants vie for Microsoft deals"; Mary Jo Foley; PC Week, 13 December 1993, pp. 57, 64.
- [144] "Spindler charts company course in light of PowerPC transition"; Andrew Grove; *MacWEEK*, 29 November 1993, pp. 1, 107.
- [145] "Apple eyes enhanced System 7"; Mike McGuire; PC Week, 6 December 1993, pp. 1, 16.
- [146] "Windows-compatible Mac due in '94"; Mike McGuire; PC Week, 22 November 1993, p. 16.

