

# Selecting an Operating System, Part I: OS/2 2.x

Stefan Zollner

*Ames Laboratory (US-DOE) and Department of Physics and Astronomy, Iowa State University, Ames, IA 50011*  
(February 10, 1994)

## I. EDITOR'S PREFACE

When I bought my first 80286 based machine (PC/AT) in 1984, I was assured by the salesman that a version of DOS which could fully exploit the capabilities of the 80286 would soon be available. Although I'm still waiting for that version of DOS, OS/2, Windows NT, NextStep, and Unix all promise to exploit the capabilities of the new 32-bit 80x86 processors. Since Stefan Zollner has been using OS/2 2.x in his lab since it became available two years ago, I asked him to prepare the first in a planned series of articles on operating system choices for computers used in the physics laboratory.

I am still looking for help in preparing articles on other operating systems. Ideas, comments, suggestions and volunteers can be sent to the editor at the email address above or via USPS at James R. Matey, David Sarnoff Research Center, CN 5300, Princeton, NJ 08543-5300.

## II. INTRODUCTION

Until recently, DOS was the operating system (OS) chosen by most users of 80x86-based personal computers (PC's). However, much more sophisticated OS's, such as Windows NT from Microsoft and OS/2 2.x from IBM, which exploit the advanced capabilities of modern PCs, are now becoming widely available. In this article, we will outline the limitations of DOS, discuss how the new OS's seek to address those limitations, and review our successes and difficulties in using OS/2 2.1 in our experimental physics laboratory.

## III. DOS AND ITS LIMITATIONS

The microcomputers I used as a college freshman in 1981 used a 5 MHz Z-80 central processing unit (CPU), with 64 kB of RAM memory and a pair of 0.25 MB floppy disk drives as mass-storage devices. Today, typical 80x86-based personal computers<sup>1</sup> have a 66 Mhz 80486DX2 CPU, internal numerical coprocessor and cache, 20 MB of RAM, and a fast hard disk with 300 MB+ of mass-storage – all at less cost than the Z-80 based system of 1981. However, despite this dramatic change, the architecture of DOS, the OS most often used to control the hardware, has remained the same.

DOS<sup>2</sup>, much like its predecessor CP/M, is a single-user, single-tasking OS. It was designed for obsolete CPU's, the 8088/8086, and does not support the advanced features of the newer processors on which it runs today. DOS offers the user a simple command shell – the C:> prompt – through which users can load executable programs, and provides access to the peripheral hardware (display, keyboard, disks, etc.) through interrupts to its BIOS (basic input-output system). As a UNIX programmer once told me, DOS is not really an OS, but merely a task loader.

DOS provides little supervision to the program and cannot intervene in case of fatal programming errors. In many cases, forcing a reboot with the three-fingered salute (pressing Control-Alt-Del) is the only way to regain control from an errant program. Since the executing program has essentially complete control of the hardware, it can do everything from sending characters to the screen to reformatting the hard drive. With such access, innocent programming errors and deliberate vandalism (viruses) can lead to expensive, unwanted effects.

The limitations of DOS hinge upon memory allocation and memory access which were determined by the 8088/8086, see Fig. 1: The 8088/8086 CPU had a 20 bit address space, capable of addressing 1 MB. DOS allocated the upper 384 kB of this space (upper memory) for the OS and the hardware (for example as video memory), and the lower 640 kB (conventional memory) to user programs, except for a small area at the bottom used for DOS. At a time when most programs ran in 64 kB, 640 kB was awesome. However, there is a corollary to Parkinson's laws<sup>3</sup>: "Computer applications will grow in size until they absorb all memory available to them". Today, many commercial computer applications such as word processors, spreadsheets and analytical tools require 512 kB or more. In addition, hardware such as mice, network adapters, and IEEE-488 require memory for their drivers/control code and memory resident software such as anti-virus programs and keyboard enhancers require still more. With all these in competition for 640

kB, it is easy to run out of memory – even though a modern computer may have many megabytes of memory above the 1 MB boundary (extended memory), which DOS cannot effectively provide to user programs.

DOS 5.0 and subsequent releases eased the memory crunch somewhat by taking advantage of unused memory addresses in the upper memory area and by using a quirk of the PC/AT hardware design to access the first 64 kB of extended memory. Memory managers such as the EMM386 help to fully utilize the upper memory space and provide some access to extended memory.

None-the-less, it is still easy to run out of memory under DOS. As an example, under DOS, we cannot run our x-ray simulation program when our network software is running. In practice, it is often necessary to build multiple system configurations (via CONFIG.SYS) and to reboot the computer with a new configuration when changing applications – we collect data under one configuration, reboot to analyse it and reboot again to startup the network to send the analysis to a colleague for comments. If the applications can be separated in this fashion, rebooting is annoying; if they cannot, you are stuck.

Two other aspects of memory allocation and access are results of the memory addressing scheme of the 8088/8086 processor. (1) The 8088/8086 was a 16-bit processor, hence DOS is a 16-bit OS which moves data two bytes at a time, even when running on a 32-bit CPU which can move four. This results in a worse case performance degradation of 2X. (2) The 8088/8086 used two 16-bit registers, called the segment and offset, to form a single 20-bit address. The segment was shifted four bits to the left and added to the offset to get the address. Within this scheme, memory is broken up into 64 kB chunks and managing code and data objects larger than 64 kB, is significantly more difficult than managing smaller objects.

#### IV. OS'S USING ADVANCED FEATURES OF THE 80386/80486 PROCESSOR

The 80386/80486 CPU's have 32-bit address and data buses which can theoretically access 4096 MB of memory,<sup>4</sup> though current machines typically provide access to 128 MB. Although the 80386/80486 can address memory using a segment:offset model, they also provide a flat, un-segmented model in which large arrays and jumps to remote locations in memory need not create problems for the OS or user program. The 80386/80486 have memory management hardware which makes it possible to protect areas of memory against unauthorized access and to map non-contiguous blocks of physical memory (chips/address lines) into contiguous blocks of logical memory (program addresses). The 80386/80486 also have hardware support for creation and control of multiple tasks. Although the 80386/80486 have other important features, the improvements in memory allocation/protection and access and the support for multi-tasking are by far the most important.

DOS runs on the 80386/80486 without taking much advantage of these capabilities. Windows 3.1 (Ref. 5), which rides on top of DOS, takes advantage of somewhat more of the capabilities in its enhanced-386 mode, however, full utilization requires a true 32-bit OS.

IBM's OS/2 2.0, released in April of 1992 was the first major<sup>6</sup> 32-bit OS for the 80386/80486. OS/2 2.1 fixed most of the problems of OS/2 2.0 and added functions including 32-bit graphics engine, support for Windows 3.1, multimedia and non-IBM hardware. At the time of this writing (Fall,1993), Microsoft had just released<sup>7</sup> Windows NT, also a 32-bit OS and is reportedly working on a 32-bit replacement for Windows 3.1, code-named Chicago, which apparently still includes the 640 kB barrier, see Table I. There are rumors that DOS 7.0 will have a 32-bit flat memory model.

We began using OS/2 2.0 shortly after it became available, in order to fully exploit the power of our 80386/80486 computers, make it easier to use the same computer for a variety of purposes, and for controlling multiple processes (or devices) within the same experiment.

#### V. OS/2 VIRTUAL MEMORY AND MULTITASKING

The two most important features of OS/2 2.x are protected, virtual memory and preemptive multi-tasking, both of which rely upon hardware support from the 80386/80486.

Under DOS, memory is real (or physical) memory. To run a program requiring 512 kB of memory, 512 kB of available, physical memory must exist in the computer. Under OS/2 2.x, each process/program is given a virtual address space of 512 MB, which looks like 512 MB of RAM to the program even if the computer only has 8 MB of memory. The OS keeps track of which parts of virtual memory are in use by the program and maps blocks of physical memory into the virtual address space. When the OS runs out of physical memory it swaps physical memory to disk by selecting a virtual memory block (belonging to the same or a different process) that is not in use, copying it to a disk file, unmapping the virtual memory block and then remapping the freed physical memory as needed. When

the memory swapped to disk is needed again, the OS locates a free physical memory block, swapping to create it if needed, and then copies the contents from the disk file and remaps the restored block to virtual memory.

Virtual memory is completely transparent to the user and greatly simplifies numerical calculations on the PC requiring large matrices. However, TANSTAAFL !! (translation – There ain't no such thing as a free lunch). Frequent swapping will seriously degrade the performance of any program. Although OS/2 2.1 will run in only 4 MB, a program accessing a large matrix might take hours to run on a 4 MB machine due to swapping, and might run in minutes on a 16 MB machine, if the matrix can fit into memory without swapping.

Virtual memory provides a large enough memory space that we can load into memory all of the device drivers, networking software, productivity tools and applications that we need without worrying about running out of memory space.

Multi-tasking takes advantage of the fact that our CPU's are frequently idle. As I sit at the keyboard writing this article, the CPU is spending most of its time waiting for the next key stroke. With multi-tasking, I can be running two or more programs at once – perhaps a numerical simulation and my word processor. The numerical simulation program can use many of the CPU cycles which are being wasted waiting for keystrokes, without having serious impact on the word processor. Multi-tasking can also provide me with the capability to easily switch back and forth between two interactive programs such as a word processor and a graphics program.

DOS and its children provide a limited version of this capability. However, since each process in DOS has access to all of the computer, an error in any of processes has the capability of trashing all others. Since few commercial, let alone user written, programs are bug free, DOS multitasking is rather fragile.

OS/2 2.1 uses the memory and process control hardware of the 80386/80486 to provide secure operating environments for multiple programs. Most parts of the OS kernel (and some low-level device drivers) run in a CPU mode called ring 0 or supervisor mode, whereas the application programs run in user mode or ring 3, see Fig. 2. The CPU was designed so that only ring 0 code (that is only the OS) can transfer control from one process to another. Each ring has its own stack (temporary storage), so an application cannot corrupt the OS stack. A number of privileged CPU commands (for example the INP and OUTP commands which directly access hardware registers) cannot be executed<sup>8</sup> under ring 3. If a program wants to access the hardware, it usually has to issue a call to a device driver. This can be a problem. If the maker of one of your interface cards, plotters, or printers, has not provided an OS/2 driver, you won't be able to use it under OS/2. You may have to learn to write an OS/2 driver – a much more difficult task than controlling the card with a few simple BASIC or assembly language statements. (There is a workaround for port I/O.)

OS/2 uses the 80386/80486 memory protection mechanisms as well. Each application is only allowed read and write access to specific areas of memory. If the application gets out of control and attempts to access a memory segment that is off limits, the CPU interrupts the process and returns control back to the OS indicating a memory protection fault has occurred.<sup>9</sup> The OS in turn informs the user that the application had to be terminated. The OS also offers the program a chance to trap the interrupt and attempt a graceful recovery. In either case, system integrity is never in danger, and other applications are not affected.

In addition to making it difficult for an errant application to crash the OS or another applications, this design provides a substantial degree of protection against computer viruses. At this time, there are a number of virus protection programs for OS/2 2.1, but no known OS/2 viruses.

Although this design seems perfect for multitasking, in practice it is not perfect. Bug free operating systems do NOT exist and OS/2 2.1 has its share of bugs. (Bug reports are called APARs by IBM. A listing can be obtained from `envy.almaden.ibm.com`.) When using OS/2 2.0, we encountered a number of cases where the OS hung and had to be rebooted. In the worst case, the OS files on the hard disk became corrupted so badly we had to reformat the partition and re-install the OS – a time-consuming process. Using true-blue IBM equipment (without any network cards, FAX boards, modems, or SCSI adapters) OS/2 2.1 is reasonably stable. At the present, adding third-party interface boards which require non-IBM device drivers jeopardize that stability. In particular, early versions of IBM's TCP/IP software for ethernet had severe problems. Fixes for many problems can be found on IBM's corrective service disks (CSD's) which are available free of charge to their customers and usually distributed over the Internet. A CSD for OS/2 2.1 (containing additional video support, non-SCSI CD-ROMS, and fixes for a few hundred APARs) is currently in beta-test and will be released shortly.

## VI. OS/2 COMMAND SHELLS AND USER INTERFACES

The bulk of a user's interaction with the operating system is via the command shell or graphical user interface (GUI). OS/2 2.1 provides multiple command shells and graphical user interfaces. As illustrated in Fig. 2 there is (1) a virtual DOS command processor (VDM), (2) an OS/2 command processor (both either full-screen or windowed), (3)

capability of booting a specific DOS version from a floppy disk image (virtual machine boot VMB), (4) WIN/OS2, OS/2's version of Windows based on the Microsoft Windows kernel, and (5) Presentation Manager (PM) the native OS/2 GUI which includes an object-oriented workplace shell (WPS). Other user interfaces are available as add-on products from third-party vendors.

After OS/2 boots, a VDM can be started by double clicking on the "DOS Full Screen" icon in the "Command Prompts" folder. The resulting virtual DOS machine has access to the hardware via a DOS virtual device driver and appears very much like a real DOS machine including an AUTOEXEC.BAT file and DOS entries in OS/2's CONFIG.SYS. The system configuration of the virtual DOS machine can be tailored by adjusting the settings notebook of the DOS Full Screen object.

Most "well behaved" DOS applications will run properly within a VDM; some require fine tuning of the DOS parameters. Others, particularly games and other programs which bypass the DOS BIOS and directly address hardware to get high performance, will fail. Since the VDM emulates an 8086 processor, DOS programs which bypass the OS to make use of 80286/80386/80486 capabilities will also fail. Information about appropriate settings of the DOS parameters, and other compatibility issues can often be found on the internet, see Table II.

If a VDM cannot handle a program, the last remedy is to run the program under a specific DOS image, either by booting from floppy or from a disk image. This still allows some access to OS/2 (for example to switch back and forth between PM and the specific DOS processor), but gives the DOS program as much access to the processor as possible. Direct DMA access is possible, but the number of interrupts is limited to about 1200 per second. Some poorly designed DOS programs will not run well under OS/2. The Wordperfect text editor (Version 6.0 for DOS) falls under this category. This program generates more interrupts to the keyboard than necessary for even the fastest typist in the world. The large number of these interrupts (from the DOS virtual device driver to the OS/2 device driver to the hardware. see Fig. 2) generates too much overhead and brings program execution to a crawl. (If OS/2 cannot run a DOS program, you can still run this program on your PC by installing the dual boot feature allowing you to switch between OS/2 and DOS by rebooting.)

The OS/2 command shell is a superset of the DOS command set. It provides commands specific to the multitasking support of OS/2 as well as extended file attributes (EA's) and long file names provided that the the HPFS (high-performance file system) has been installed on the hard disk. For floppy drives, OS/2 uses the same FAT (file allocation table) file system as DOS. When a DOS executable is invoked, the OS/2 shell automatically starts a DOS subprocess to run the program. This is somewhat inconvenient, since the screen is cleared before and after opening the DOS box, erasing all program output.

OS/2 command files (CMD) are much more powerful than DOS batch files because OS/2 provides a REXX interpreter for the CMD files. REXX has a C-like syntax, many OS-specific functions, and is similar to the REXX found on IBM mainframes.

## VII. GRAPHICAL USER INTERFACES

OS/2's presentation manager (PM) and the workplace shell (WPS) provide an object-oriented Macintosh-like interface (see Fig. 3), with much of the power of a UNIX workstation. The mouse can be used to select, move and open objects on the graphical desktop. Documents can be associated with applications on the basis of file name extension (e.g. .TEX) so that double-clicking on a document icon will start the appropriate application. Dragging an object on top of another object and dropping it there initiates an appropriate action. Dropping a document on a folder moves the document to the folder; dropping it onto the shredder deletes it; dropping it on an application icon starts the application and loads the document into the application. The window for a sample PM application along with its controls is shown in Fig. 4.

OS/2 2.1 includes a GUI which can run most 16-bit Windows applications under OS/2, either seamless (in a window on the OS/2 desktop) or full-screen. Although OS/2 2.0 had a number of problems in running Windows programs, OS/2 2.1 does much better, in part because it uses code licensed from Microsoft. It therefore runs Windows programs much better and also provides support for Windows 3.1 programs. However, most vendors of Windows software do not focus on OS/2 2.1; many of them do not test their software with OS/2 2.1 and do not promise OS/2 compatibility. In most cases, there are no problems, but there are peculiar exceptions: Most Claris software, for example, will not print from OS/2 2.1 because of an uninitialized variable in the code which only affects OS/2 and not Windows 3.1. The vendor is not moving rapidly to fix the problem because everything is fine in Windows 3.1, and IBM is not reacting because it is a bug in the application.

## VIII. OS/2 APPLICATION PROGRAMS

The most common criticism of OS/2 2.1 is the lack of native applications. It is true that few native OS/2 applications<sup>10</sup> are available at this time, however, most DOS and Windows applications will run under OS/2 2.1 and many public-domain UNIX utilities, such as EMACS and the GNU programming environment, have been ported to OS/2. Many freeware and shareware programs for OS/2 or WIN-OS/2 can be retrieved via anonymous FTP from software depositories listed in Table II.

A number of applets (small applications) are included with OS/2 2.1. Though not formally supported by IBM, they demonstrate the capabilities of the OS and, in some cases, are demo versions of third-vendor packages. Some of these applets are so excellent that I use them on an almost daily basis: There is a personal planner package, complete with calendar, alarm, and to-do-list, the enhanced editor EPM, which can be configured and recompiled by the user, and PMCHART, a mini-graphics program, which I use to prepare drawings; some of the figures in this paper were drawn with PMCHART.

Our most important single reason for switching to OS/2 was its networking capabilities. As noted earlier, it can be difficult to run networking software simultaneously with large applications under DOS. The networking protocol most commonly used in a multi-vendor university environment is TCP/IP, originally developed for the UNIX world. The hardware adapter to run this protocol is usually an Ethernet or Token-Ring adapter. These adapters are available for PC's from a variety of vendors. I attempted to use TCP/IP programs under DOS, but found that the drivers left me with too little memory to run my applications programs particularly after loading Windows 3.1, see Fig. 1.

Under OS/2 2.1, I have all the features of TCP/IP, including TELNET, FTP, GOPHER, X-Windows, and NFS client and server, E-MAIL, and news reader. Because of the flat-memory model, there are no memory conflicts. I do, however, recommend 16 MB of RAM to avoid too much swapping. With TCP/IP and its integrated networking support, OS/2 2.1 becomes an excellent choice as the OS for a client-server environment on both the client and the server machines.

Here is a list of my favorite OS/2 applications. Not all of these are OS/2-native applications, but they run well on OS/2 2.1:

- IBM TCP/IP 2.0 for OS/2 (including TCP/IP base, e-mail, NFS, DOS/Windows access, and X-Windows server).
- IBM C/C++ compiler tools, including C/C++ compiler, PM-oriented debugger, Workframe/2, an integrated development environment (IDE), linker, tool kit, and on-line programming library.
- Borland C/C++ compiler, linker, IDE, and resource development kit. The performance of the current release is disappointing, but the design looks promising. A maintenance refresh has been promised for almost a year.
- National Instruments IEEE bus drivers (NI-488.2M) for OS/2 2.1.
- Quicken for Windows (Intuit). Financial planning program.
- GNUPLOT, a simple command-line oriented program for  $x - y$  charts.

## IX. HARDWARE REQUIREMENTS AND SETTING UP OS/2 2.1

OS/2 2.1 will run on any modern PC with a 80386SX CPU or better and at least 4 MB of RAM. However, I recommend at least an 80486DX33 processor and a minimum<sup>11</sup> of 8 MB for a stand-alone or portable PC. Other features, such as networking software, the C++ compiler, seamless WIN-OS/2, high-resolution SVGA, the HPFS file system, and multimedia require even more RAM, but 16 MB should generally be sufficient. The additional memory dramatically improves performance by reducing swapping.

An ISA-bus machine will be less expensive than EISA, but I recommend at least two VESA local bus slots for the hard disk and video adapter cards to provide better data transfer rates. IDE hard drives avoid the SCSI incompatibility problems often encountered on the PC. Get at least 200 MB (more if you can afford it) and an OS/2 compatible SCSI tape drive, unless you can conveniently backup over a network.

At the present time, OS/2 drivers are not available for many of the advanced video cards. Hence, your best bet may be to use an inexpensive SVGA card with 1 MB of video RAM on the board. In any case, make sure that your video card comes with drivers for OS/2 2.1 in 1024×768 resolution and 256 colors (including seamless WIN-OS/2 support) or is supported by IBM's drivers.

A configuration like this can be bought for \$ 1500 to 2000 (September 1993). I recommend buying an inexpensive machine with a one-year warranty. After a year, the machine is obsolete and has lost most of its value; if a component fails, it can usually be replaced (at least by an experimentalist).

Optional items include a network card, SCSI adapter, CD-ROM player, fax board/modem, interface cards to control your experiment (IEEE-488, analog-to-digital converter, indexer card for stepping-motor control, digital I/O) and a parallel laser printer supporting various formats (HP Laserjet PCL, HPGL graphics language, Postscript, IBM PPDS)

in at least 300 DPI resolution. With a large number of cards, you may encounter interrupt conflicts with an ISA-bus machine, since cards cannot share interrupt levels. You also may need an expansion chassis if you have more cards than slots in your PC.

OS/2 2.1 is available in three versions: On floppy disks (at least 18), on CD-ROM, or as a license without media. If there is already a license for DOS and Windows 3.1 on the machine, then the best-selling OS/2 upgrade with codename Ferengi (also called OS/2 2.1 for Windows) can be purchased starting at about \$30. Installing from floppies may be the easiest way (no problems with unsupported SCSI adapters or CD-ROM players), but using a CD-ROM is much faster. The installation is usually without complications, but special cases may require some planning, or hunting for drivers on the internet. Upgrading from Windows 3.1 or OS/2 2.0 (with or without the service pack) is also painless. It is not necessary to reformat the partition (except when upgrading from a previous beta-test release), and your previous desktop will be preserved.

## X. SUMMARY

We have been studying the feasibility of OS/2 2.x in our laboratory for the past eighteen months. We found that OS/2 2.1 is more difficult to install and maintain than DOS (together with Windows 3.1) and that it does not adequately support all hardware we have tested. OS/2 2.1 and its applications also contain a number of bugs that have to be worked around. Nevertheless, we chose OS/2 2.1 as the platform for our experiment (an ultrafast pump-probe rotating-analyzer ellipsometer with femtosecond time resolution) because its superior design and multiprocessing support make it easier to control complex tasks.

OS/2 2.1 provides us with an environment in which we have access to all the tools we need for our work without constant rebooting of the OS and in which we need not fear the effects of errant applications on our experiments. It does require more resources than DOS, but the incremental cost for those resources is now so low that the productivity improvements we get from a modern operating system are well worth the cost.

## XI. EDITOR'S POSTSCRIPT

Working with Stefan on this article convinced me to give OS/2 a try. I bought the \$40, OS/2 for Windows upgrade and installed it in dual boot mode on one of my DOS/Windows machines. The 2 hour installation from diskettes was painless, though I did have to reprogram my 1.44 MB floppy for drive A and run Setup to reconfigure the CMOS parameter memory.

I also read an excellent book, *The Design of OS/2* by H. M. Deitel and M.S. Kogan (Addison-Wesley, 1992), which helped me understand OS/2 much better than I would from simply reading the manuals.

So far, everything except my AppleTalk network software is working. That may be a killer for me, since we run a mixture of Macintoshes and DOS/Windows machines and interconnect them with AppleTalk. I also found that OS/2 slowed to a crawl due to disk swapping when I lost a memory chip and dropped from 8MB to 4MB. I am looking forward to trying out some large image processing codes as soon as the memory is restored.

My thanks to Steve Lidard for help in editing this manuscript.

---

<sup>1</sup> In this article, PC denotes any IBM-PC or compatible computer, regardless of hardware details.

<sup>2</sup> In this article, DOS denotes the family of operating system which include MS-DOS, PC-DOS and their derivatives.

<sup>3</sup> "work expands to fill the time available for its completion". C. Northcote Parkinson in *Parkinson's Law and Other Studies in Administration* (Houghton-Mifflin, 1957).

<sup>4</sup> The CPU can also access a hard disk partition with up to 4096 MB in a single step, by putting a 32-bit address on the bus. However, that may soon be a limitation, since multi-gigabyte SCSI hard disks are already available.

<sup>5</sup> In this article, Windows denotes Microsoft Windows 3.x, Windows for Workgroups, and earlier versions of this software. This should not be confused with Microsoft Windows NT, an entirely different OS; WIN-OS/2, the Windows 3.x-compatible subsystem of OS/2 2.x; or other windowing systems such as Motif, OpenWindows and Macintosh.

<sup>6</sup> Though 32-bit UNIX for PC's exists, so far, no PC-UNIX has been able to gain a dominant market share. It is possible (though unlikely, in my opinion) that UNIX may dominate the PC OS market in the future.

- <sup>7</sup> There is no doubt that its internal architecture makes Windows NT a more advanced operating system than OS/2 2.x, at least in theory. The deserialized GUI message queue, superior handling of interrupts and time slices for multitasking, multiple platform support for Intel and DEC Alpha chips, multi-processor support, and a more secure network file system are currently included in Windows NT, but only in the testing or planning stage for future IBM OSs. In practice, OS/2 has five years of experience and is therefore more reliable. By the time NT is available in a “production” version, these features will probably also be available from IBM. OS/2 excels with its object-oriented workplace shell (making Windows Program Manager look obsolete) and better backward compatibility for DOS and Windows applications. Recent issues of the journal *OS/2 Professional* discuss these issues in more detail. We note that sales figures for OS/2 2.x have exceeded those of Windows NT (and all PC UNIX flavors) by orders of magnitude.
- <sup>8</sup> My students and I recently wrote a program for stepper motor control under OS/2 2.1, where we directly wanted to access the indexer card using INP and OUTF statements, just like we would have done in DOS. The way this can be achieved without too much difficulty is to link a 16-bit code segment running in ring 2 with the application program, where some privileged statements are allowed. Details will be discussed elsewhere.
- <sup>9</sup> The OS/2 error message SYS3175 is frequently encountered and indicates a general protection fault (access violation). This error is caused by the application, not the OS, therefore the vendor or developer should be contacted for a fix. Unfortunately, not even IBM’s tech support is responsive in fixing these errors caused by IBM applications (such as the EPM editor, IPMD debugger, or LAMAIL).
- <sup>10</sup> OS/2-native applications include the following: Wordperfect 5.2, Stacker (FAT only), DCF/2 (disk compression for HPFS), SAS (data base and spreadsheet), PFS:Works, FAX/PM, FaxWorks, CorelDraw, Borland ObjectVision, Visual REXX, Freelance Graphics, Lotus 1-2-3, Lotus Notes, GammaTech disk utilities, DeScribe (word processor), Ultimedia/2. Not all of these are well tested and worth buying.
- <sup>11</sup> If you cannot afford 16 MB of RAM, make at least sure you can later upgrade to 16 MB without having to discard your 1 MB SIMMs.

FIG. 1. Memory contents of a PC with 20 MB RAM when run under DOS 5.0 or above.

FIG. 2. Structure of OS/2 2.1 and its various components and sub-systems. The solid line separates code running in user (ring 3) and supervisor (ring 0) mode.

FIG. 3. A sample OS/2 desktop showing the various features of the workplace shell and some PM applications, i.e., printer icons for various formats (emulated by a single IBM 4029-022 printer with automatic switching), information folder, minimized icon viewer, shredder (trashcan), OS/2 system, command prompts, and programming folders, OS/2 shell window, and electronic mail application.

FIG. 4. A sample Presentation Manager (PM) window illustrating the various programming controls. (Copyright by IBM. All rights reserved. Reprinted with permission.)

TABLE I. Comparison of different operating systems for 80x86 CPU's.

	DOS 6.0	Windows 3.1	Chicago	Windows NT	OS/2 2.1	SUNSoft UNIX
upgrade price (USD)	0	0	N/A	263	39	708
32-bit OS	NO	NO	YES	YES	YES	YES
640 kB barrier	YES	YES	YES	NO	NO	NO
flat memory model	NO	NO	NO	YES	YES	YES
virtual memory support	NO	YES	YES	YES	YES	YES
graphical user interface	NO	YES	YES	YES	YES	YES
object oriented shell	NO	NO	NO	NO	YES	NO
DOS device drivers loadable	YES	YES	YES	NO	YES	??
memory protection	NO	NO	NO	YES	YES	YES
preemptive multitasking	NO	NO	NO	YES	YES	YES
deserialized message queue	NO	NO	NO	YES	NO	N/A
fixed time slice	YES	YES	YES	NO	YES	??
multiprocessor support	NO	NO	NO	YES	soon	??
non-Intel platform support	NO	NO	NO	YES	NO	??
installable file system	NO	NO	?	YES	YES	YES
recommended CPU (or better)	8088	80286	80386SX	80486	80386	80386
recommended minimum memory	1 MB	4 MB	4 MB	16 MB	8 MB	8 MB
recommended minimum disk space	2 MB	10 MB	10 MB	200 MB	100 MB	200 MB
client-server support	NO	limited	limited	YES	YES	YES
best purpose	standalone	client	client	server	any	server
DOS subsystem	N/A	YES	YES	YES	YES	YES
Windows 3.1 subsystem	N/A	YES	YES	YES	YES	NO

TABLE II. Internet and toll-free sources for OS/2 support and software.

Source	protocol/type	contents
your news server	newsgroup	comp.os.os2.* hierarchy
your news server	newsgroup	bit.listserv.os2-l
os2-l@hearn.nic.surfnet.nl	mailing list	contact listserv@hearn.nic.surfnet.nl to subscribe
envy.almaden.ibm.com	gopher	IBM information about OS/2
ftp.cdrom.com	ftp	OS/2 software, fixes, information. Start with /pub/os2/00index.txt and /pub/os2/all/info/faq/faq21d.zip
(800) 786-9907	telephone	Walnut Creek: contents of ftp.cdrom.com on CD-ROM
software.watson.ibm.com	ftp	free IBM software, OS/2 fixes, drivers
ftp.lexmark.com	ftp	IBM Lexmark printer drivers
ftp.3com.com	ftp	3COM device drivers
watson.cc.columbia.edu	ftp	kermit/bin/ckoker.zip
extreme.cica.indiana.edu	ftp	large windows archive
ftp.uni-stuttgart.de	ftp	EMTEX: $\LaTeX$ for OS/2
clio.rz.uni-duesseldorf.de	ftp	hp2xx, an HPGL graphics viewer
cset2@vnet.ibm.com	email	IBM support for C/C++ compiler tools
workframe@vnet.ibm.com	email	IBM support for workframe development environment
76711.610@compuserve.com	email	IBM tech support for OS/2 base and TCP/IP
(800) 992-4777	telephone	IBM OS/2 2.1 tech support (US only).
zollner.ssp.ameslab.gov	ftp	this article and other OS/2 information.
(800) 776-8284	vendor	Indelible Blue: specialized on OS/2 products (books, software).
(301) 770-4672	magazine	OS/2 Professional.