OS/2's History and Purpose

OS/2 was originally developed jointly by IBM and Microsoft as a multitasking successor to DOS for 286 and better CPUs, but version 1.x never really caught on except in a few specialized applications. With version 2.0, Microsoft dropped out of the OS/2 partnership, and IBM promoted OS/2 to a 32-bit OS requiring a 386 or better CPU. This basic configuration has not changed with OS/2 2.1 or 3.0. The upcoming "Merlin" version of OS/2 (probably to be called 4.0 when it's released) will not be tested with 386 CPUs, and at this point it's unknown whether it will work on a 386 system at all.

OS/2 Warp 3.0 is a multitasking, 32-bit, single-user OS for 386SX and better CPUs with 4MB or more of RAM. It is very DOS-like in some ways (such as the commands used in its command-line interface, and the presence of a CONFIG.SYS file), but resembles the Mac in other ways (e.g., the iconic representation of files from the WorkPlace Shell) and has some similarities to other OSes in still other ways (e.g., pop-up menus when clicking on the desktop itself, which are reminiscent of X Windows under Unix). Warp includes a Graphical User Interface (GUI) known as Presentation Manager (PM), and a desktop metaphor for launching programs and manipulating files called the WorkPlace Shell (WPS). The PM bears some resemblance to Windows, though it's not identical. The WPS is similar to Windows 95's desktop metaphor or the Mac's Finder, but is generally more flexible and more object-oriented than either. A Windows version of the WPS is available.

OS/2 Warp 3.0 comes in several versions with varying levels of networking support. The first-released version has only dial-up networking (SLIP and PPP) support. Two other versions, "Warp Connect" and "Warp Server," include client and server capabilities, respectively. The next release of OS/2, code-named "Merlin," will include client networking features in the base package. When used with Windows for Workgroups 3.1, the WfW networking features are disabled under OS/2, though they can still be used if WfW is run from native DOS.

The lower-end versions of OS/2 also come in two versions each, one that includes a re-compiled version of Windows (often called the "with Win-OS/2" or "Blue Spine" version), the other of which requires that the user already have Windows 3.1 in order to run Windows programs from OS/2 (somtimes referred to as the "for Windows" or "Red Spine" version, though neither term is official). The "with Win-OS/2" version sometimes runs Windows programs slightly faster than does the "for Windows" version, and is easier to set up if the user doesn't already have Windows 3.1 installed; but the "for Windows" version is less expensive.

Out of the box, Warp can run OS/2 text-mode, OS/2 GUI, and DOS programs. Windows is a DOS program which Warp can run, and this is how OS/2 provides Windows support -- by running Windows on top of its DOS mode. This method of Windows support will not change with Merlin. Note that Windows 95 CANNOT be used to provide Windows program support for OS/2.

Most new OS/2 users should get the original "for Windows" version of Warp. The "for Windows" version is slightly less expensive and will use less disk space than the "fullpack" version. Somebody who's upgrading from OS/2 2.1 fullpack should buy the upgrade package of the Warp "fullpack," which includes a "sniffer" to detect the old 2.1 code, and won't install if it doesn't find this. Somebody who's building a new computer and who doesn't already have Windows or OS/2 2.x, but who wants to run Windows programs, should buy the non-upgrade "fullpack" version of OS/2, which is the more expensive version, but more convenient than buying the "for Windows" version and a separate copy of Windows.

Major OS Features

- Multitasking. All OSes can be classified according to how they run multiple programs. At one end are simple OSes that can only run a single program at once, or which, at best, place extreme restrictions on running multiple programs. MS-DOS is an example of such an OS. One step up from this level is an OS that does "co-operative multi- tasking," in which the programs involved must voluntarily switch back and forth. Such a system typically multitasks as well as its least cooperative program; one program can easily "hog" the CPU, preventing other programs from running. Windows 3.1 and the MacOS both use co-operative multitasking. The next level up is "pre-emptive multitasking," and this lets the OS dole out CPU time as it sees fit, thus reducing or preventing problems due to one program's selfish hogging of resources. Pre-emptive multitasking allows, for instance, a reliable high-speed file transfer, spreadsheet recalculation, and floppy disk format to all occur at the same time. Windows 95, Windows NT, most varieties of Unix, and OS/2 all use pre-emptive multitasking, though there are variations in implementation details, particularly concerning the multitasking of DOS and Windows 3.1 programs. Some OSes, including some but not all varieties of OS/2, can also handle symmetric multi-processing (SMP) computers, in which more than one CPU is present. In these systems, one program can run on one CPU while another runs on another.

- Multithreading. This concept is similar to multitasking, but involves only a single program. In an OS which allows multithreading, a single program can create multiple "threads" of execution in order to improve responsiveness. For instance, a word processor might create a thread for printing a document and then allow you to immediately return to editing the document even while the word processor is still formatting the text for the printer. Note that the OS ENABLES multithreading, but to be useful this feature requires applications that utilize the feature. OS/2, Windows 95, and Windows NT all enable multithreading for native applications, but not for "legacy" Windows 3.1 or DOS programs (unless these programs implement the feature internally).
- Process protection. In some OSes (such as Windows 3.1), one program can write into memory reserved for another program, or sometimes even the OS, thus crashing the other program or the OS as a whole. OS/2, as well as Windows 95, Windows NT, and most Unixes, includes a method to prevent this from happening. This system is imperfect, but it does work most of the time. OS/2 extends this protection to DOS programs, and can prevent Windows programs from damaging DOS or OS/2 programs. Normally, Windows programs can still damage each other under OS/2, but by running Windows programs in "separate sessions," this protection can be extended to them, as well. Windows NT's level of process protection is roughly comparable to OS/2's, while Windows 95's is comparable only for native Windows 95 programs; it's my understanding that Windows 3.1 and DOS programs run under Windows 95 can still do considerable damage much more easily than they can under OS/2.
- The "64kB limit." Because of its 32-bit address space, OS/2 programmers needn't worry about the so-called "64kB limit" imposed by DOS's history. Under DOS, a single data structure is limited to 64K by the 16-bit memory addresses used by DOS, unless a 32-bit DOS extender is used. This 64K limit can be a pain to programmers, though end users needn't ordinarily worry about it. In fact, "32-bit" has been over-hyped by the media. Although some programs may get a speed boost from being 32-bit, others may win a speed *decrement* if re-coded as 32-bit. Thus, the 32-bit address space is mostly a concern for those doing programming using large data structures. It should be noted that the new Intel Pentium Pro processor wins a speed increment for 32-bit code but a decrement for 16-bit code, compared to a similarly-clocked Pentium processor. Thus, OS/2 users who run mostly native 32-bit OS/2 programs can benefit from the Pentium Pro, but people running DOS/Windows, or even Windows 95 (which contains a substantial amount of 16-bit code internally) may not. Windows NT and most Unixes are mostly or completely 32-bit internally and run 32-bit native programs.
- The "640kB limit." OS/2's memory architecture also obliterates the so-called "640kB barrier" imposed by DOS. While DOS programs are still limited to less than 1MB of RAM directly (plus whatever EMS, XMS, or DPMI memory they need), OS/2 itself is not so limited. The main upshot of this is that an OS/2 user need not juggle device drivers endlessly to maximize the available main memory. For many devices (such as CD-ROMs), OS/2 gives DOS programs access to the device without explicitly loading a DOS device driver, assuming that an OS/2 device driver is used. Windows 95 is a bit of a cross on this issue; when using native drivers, the 640kB barrier isn't much of an issue, but when using older DOS drivers it is. Windows NT and most Unixes are pretty much immune from the 640kB limit, as well.
- DOS settings. Many DOS and Windows programs require special drivers or other commands in CONFIG.SYS or AUTOEXEC.BAT, and these commands or drivers sometimes interfere with other programs, creating a problem for the user. OS/2 allows the user to specify a unique AUTOEXEC.BAT file and assorted settings that influence the way OS/2 runs the program, thus giving the user great flexibility in running individual DOS programs, and even in running several programs with conflicting requirements simultaneously. Windows 95, Windows NT, and Unixes (when they support DOS programs at all), by contrast, are much more limiting in this respect. Windows 95 may produce better results in running some finicky DOS applications individually by shutting itself down into a dedicated DOS mode,

but in terms of customizing settings for specific DOS programs, OS/2 is the most flexible OS available.

- Filesystems. Hard disks must be organized using a disk-spanning data structure known as a "filesystem," which allows the OS to locate specific files. The filesystem used will have an important impact upon system performance and features -- the filesystem determines, for instance, the rules for naming files and the granularity of disk space allocation (e.g., is disk space allocated in 0.5kB chunks, 2kB chunks, etc.). To most people, FAT is the most familar filesystem, but it isn't a very good one; it includes too many limitations and design problems, some obvious to the user (like the so-called "8.3 filename limit") and others not (like increasingly large disk space allocation units with increasing partition size, which can end up wasting 30% of a 1GB or larger disk's available space). While OS/2 can use FAT, it includes the optional HPFS filesystem, which offers long filenames, a constant 0.5kB allocation block size, better speed, and greater reliability relative to FAT. DOS and Windows programs can run on HPFS partitions from OS/2, but will still suffer from the 8.3 filename conventions of DOS. While switching from FAT to HPFS isn't necessary for OS/2 users, such a switch is often quite beneficial. Other OSes offer other filesystems; for instance, Windows NT offers NTFS, which is quite similar to HPFS in many ways; and Linux uses ext2fs. Windows 95 uses something called VFAT, which is basically FAT with long filename support. Recent systems with Windows 95 preloaded may use VFAT-32, which offers relatively small 2kB allocation block sizes on most hard disks.
- Networking. Most OSes these days offer some form of networking support. As described above, OS/2 Warp 3.0 comes in different versions with varying levels of network support, though Merlin is slated to simplify matters considerably, bundling client networking capabilities in the base OS.
- Multimedia. Most OSes today can handle multimedia applications ranging from simple sound playback and recording to full-motion "movies" and dynamic 3-D displays. OS/2 provides the necessary underlying OS components for multimedia support, in the form of MMPM/2. Aside from playing sounds, however, there are few OS/2 programs that utilize multimedia in any substantial way, though this is slowly changing. Most Windows multimedia titles will run under OS/2, though some (particularly those which use MIDI file playback) may produce somewhat "jerky" timing. Integrating the Windows sound card drivers is the weakest point in the OS/2 multimedia link. Because most of the home multimedia titles are aimed at the masses, these titles do work best under Windows 3.1 or Windows 95. Some Unixes do provide native multimedia support, but these OSes generally don't run Windows multimedia titles very well, if at all.
- Desktop Shell. OS/2's WPS provides a unified method of access to programs and files, something like the Mac's Finder. The WPS is very flexible and extensible, both by the user and by programmers. The main drawback to this flexibility is that it increases system overhead, slowing down certain operations. Windows 95 includes a similar, but not as flexible, user interface, and Windows NT 4.0 introduces this interface for NT. Unix X Windows servers vary considerably in their details.
- Voice recognition. The vast majority of computer users today give commands to their computers primarily via the keyboard and/or mouse. Emerging technologies are making it possible to use the human voice to accomplish this task, however, as well as to dictate directly into a computer application. While such features can be added to many OSes today, the next version of OS/2 (code-named "Merlin" and due out in late September, the last I'd heard) will include it in the base OS. No matter what OS is used, however, speech recognition will require either specialized add-on hardware or a fairly high- end computer. Merlin will require at least a Pentium-90 with 24MB of RAM, for instance, to accomplish voice dictation, with lesser requirements for voice commands.

These, then, are some major OS features, and where OS/2 fits in on them. As you have probably gathered from this, modern OSes are more alike than unlike on most of these features. For a more direct comparison of OS/2 to Windows 95, see that section. If OS/2 sounds appealing by these measures, then score a point for the "yes" vote to "should I buy OS/2?". Somebody who just uses a computer to run, say, a word processor, and absolutely nothing else, is less likely to be drawn to OS/2, though, since these features offer relatively few advantages to such a person. Somebody who wants pre-emptive multitasking, or who runs lots of DOS programs that require conflicting configurations, may find OS/2 tantalizing. That may not be reason enough to buy it, though....

This is the *theory* behind OS/2's design. The *practice* is often less rosy, since there can be device driver conflicts, incompatible hardware, and a considerable learning curve in setting up an OS/2 system optimally. For whatever reason, some people find OS/2's promises of crash protection, better multitasking performance, or whatnot to be only partially fulfilled. This is sometimes the result of flakey or incompatible hardware, sometimes the result of a non-optimal configuration, and sometimes the result of bugs within OS/2. Unfortunately, the only way to know how well OS/2 will work for you is to try it, though at least some hardware problems can be caught before picking up the box (see the hardware section). Configuration problems can often be worked out by getting help on the net. In this respect, somebody who's unwilling to take some time optimizing and possibly debugging a system should probably avoid OS/2 -- though such a person should also avoid DOS/Windows or Windows 95 if s/he is new to the PC world, since these configurations can be as difficult to a PC newbie. Such a person should view Unix as the bubonic plague.

It should be noted here that occasional incompatibilities or driver bugs will prevent OS/2 from even installing correctly. In many cases, these problems can be overcome by obtaining updated drivers. These are generally posted on the Cincinnati Team OS/2 Update List (the first choice for this), the IBM Europe device driver repository, software.watson.ibm.com, ftp-os2.nmsu.edu, and ftp-os2.cdrom.com, as well as various BBSes. Driver problems have been reported with IBMKBD.SYS, IBM1S506.ADD, just about every SCSI driver known, and others. If you press while the white square and "OS/2" are visible in the upper left corner of the screen when booting the OS/2 installation disk, OS/2 will print the name of the driver it's loading, so you can track a driver that simply hangs your system. It's important to note, however, that while problems *exist* with many drivers, they're *uncommon* with most drivers.

Go to the next section
Return to the "Should I Buy OS/2" main page