PLAINTIFF'S EXHIBIT 826 Comes v. Microsoft

```
American competition. Saw it as a mature
action.

FYI

Peter

From brianmo Fri Jul 12 18:19:49 1991
To: isvmkt
Subject: OS/2 PM apps for SWATting etc.
Date: Wed Apr 29 17:43:16 PDT 1992

Date: Fri Jul 12 18:28:09 PDT 1991

try \\isv_server\public\user\brianmo\select\pmapps.xls
There are almost 400 of these things.
-bkm

From alistair Wed Jul 17 17:34:43 1991
To: janw lindag
Cc: markry sysmkt
Subject: Re: Windows v. OS/2 (long)
Date: Wed Apr 29 17:43:16 PDT 1992

Date: Wed, 17 Jul 91 17:33:47 PDT


Please will you arrange that the article which you once wrote as below
is removed from online - it no longer reflects our opinions or
the future of either windows or os2.

thanks, Alistair - Systems Marketing

|>From markry Wed Jul 17 17:06:21 1991
|To: sysmkt
|Subject: Windows v. OS/2 (long)
|Cc: bens markcl samjad stconly
|Date: Wed Jul 17 20:04:46 1991
|
|
|The following is a live article in the Microsoft Knowledge
|Base ... my oh my how times do change....
|
|==================
|
| Comparison of Windows 3.00 and OS/2                    [O_Os2PrTk]
| ID: Q64480     CREATED:  2-AUG-1990   MODIFIED: 19-SEP-1990
! 1.20
| OS/2
```

EXHIBIT 1 HORAN 10-25-01

ENDUSER | SR# G900601-96 O_OS2SDK

Question:

Microsoft's Windows version 3.00 seems to offer many of the features of OS/2 at less cost. Consequently, my management has asked me to justify our company's continued involvement with OS/2. What are the benefits of OS/2 over Windows?

Response:

There are three basic areas a company should consider when choosing between OS/2 and Windows: features, system requirements, and product direction.

THE FEATURES PERSPECTIVE

OS/2 provides specific features that allow much more sophisticated applications to be written than can be created under Windows. These features fall principally into the five categories of multitasking, memory protection, interprocess communication, installable file systems, and connectivity.

Multitasking

In the category of multitasking, Microsoft Windows is a real ground-breaker for users of MS-DOS. MS-DOS was originally designed to run just one program at a time. This design limitation forces MS-DOS users to terminate their currently running application before they can use a different application. Microsoft Windows frees MS-DOS users of this design limitation by providing them with a standard GUI (graphical user interface) for the MS-DOS operating system that enables them to run more than one Windows application almost simultaneously.

The Windows GUI brings multitasking to the MS-DOS world by employing a nonpreemptive, message-based scheduler. In simplest terms, this scheduler lets a Windows application have control of the CPU as long as that application has some work it needs to do. More specifically, each application keeps a list (or queue) of messages that describe things (or events) to which that application must respond to. As long as the currently executing application has a non-empty message queue, that application maintains control of the CPU; all other Windows applications are blocked. Only when the currently executing application exhausts its message queue (that is, runs out of work to do) will Windows transfer control of the CPU to another Windows application.

While Microsoft Windows has done amazing things to bring multitasking to MS-DOS, Windows' lack of a preemptive multitasking scheduler presents a problem for those Windows applications that need to perform fairly lengthy operations. Because only one Windows application can be executing at a time, the currently executing Windows program must quickly process the messages in its message queue. If the currently executing program does not yield control in a timely fashion, all other Windows applications will remain blocked until that program finishes its processing. As a result, Windows application developers must take long, time-consuming operations and break them down into small, discrete units so that the application can frequently yield control of the CPU to other applications. This lengthy operation then must wait to continue its operation until the other Windows applications have processed the messages in their message queues. This problem is a burden not only to the Windows developer (who must break down his or her program into artificially small chunks), but also to the Windows user, who must endure the added delay involved in performing the lengthy operation.

The OS/2 Presentation Manager (PM) shares Windows' message-based architecture, but because PM is an integral part of OS/2 -- a fully preemptive, multithreaded, multitasking operating system -- PM is able to deal with the problem of lengthy operations in a manner that is more natural for developers of PM applications, as well as more responsive to users of PM applications.

OS/2's preemptive multitasking scheduler has the ability to take control of the CPU away from an application and switch it from task to task very rapidly -- up to 32 times a second. In addition to scheduling the execution of multiple, concurrent applications, OS/2 provides for multiple, simultaneous "threads of execution" within a single application. In other words, not only does OS/2 allow more than one program to execute at once, it allows any given OS/2 program to execute more than one thread of execution at the same time. Consequently, an individual OS/2 application can perform multiple foreground and background tasks all at the same time.

PM applications can take advantage of OS/2's sophisticated multitasking features to provide an effective solution to the problem that Windows applications have with performing time consuming operations. Consider the case of a spreadsheet that has to perform a lengthy recalculation. A PM spreadsheet could address this problem by using two separate threads of execution: one thread could be dedicated to recalculating the spreadsheet, and the other thread could be dedicated to processing user input. Then, when the user requests a potentially time consuming recalculation, the PM spreadsheet could start the recalculation as a background thread of execution, while the main thread continues to respond to user input.

As a consequence, the PM application does not have to wait for the calculation to finish before it can accept further user input, and other PM applications can attend to their message queues while the spreadsheet is waiting for user input.

In summary, using the multithreaded capabilities of OS/2, PM applications can allow multiple operations to overlap, making the program faster, more responsive, and more efficient. Moreover, PM developers do not have to worry about chopping up time consuming operations into small, artificial chunks in order to maintain the overall responsiveness of their application.

Memory Protection

OS/2 applications are protected from corrupting each others' memory. That is, an OS/2 process cannot inadvertently overwrite the data of another OS/2 process. OS/2 uses specific features of the "protected" mode of the Intel 80286 and 80386 microprocessors to protect memory between processes. This provides the same level of reliability and integrity that is typically found on mainframe computers.

While Windows applications run in protected mode of the 80286 or 80386, they are not given this same level of protection. A misbehaved Windows application can write over the data of any other Windows application.

Interprocess Communication (IPC)

OS/2 provides the application programmer a sophisticated set of system services that lets concurrently executing programs share data, serialize access to critical resources, and exchange messages. In particular, the OS/2 kernel supports shared memory, anonymous pipes, named pipes, queues, and semaphores. In addition, the PM GUI gives PM applications the ability to deliver messages to each other. This ability to exchange messages between applications is supplemented by the DDE (dynamic data exchange) protocol -- a standard protocol for defining the meaning of messages passed between PM applications.

Currently, Microsoft Windows only supports inter-application message exchange and the DDE protocol.

Installable File Systems (IFSs)

OS/2 provides users and system administrators the ability to install custom file systems in addition to or in place of the standard

MS-DOS FAT (file allocation table) system. This new technology forms the foundation for OS/2 to be able to read files from custom network file systems, database-optimized file systems, or even UNIX or Macintosh file systems, once some party chooses to develop IFSs to support these types of file systems.

Microsoft has already capitalized on this technology by providing the new HPFS (high performance file system) in OS/2 versions 1.20 and later. HPFS is an installable OS/2 file system that is two to ten times faster than the MS-DOS FAT file system, supports long filenames (up to 255 characters -- including mixed case and blank characters), and introduces EAs (extended attributes) for files. The big win for OS/2 users here is that HPFS provides improved performance with EXISTING applications, plus new file naming flexibility. In addition, EAs give users the ability to store in their documents notes, historical data, keywords, and associations to particular applications, making the system much more flexible and easier to use. Windows does not support IFSs.

Connectivity

OS/2's multitasking and multithreading features lie at the core of Microsoft OS/2 connectivity products: the OS/2 LAN Manager local area network, the Comm Server wide area networking product, and the SQL Server database product. Microsoft's connectivity products provide advanced networking and superior client-server support for OS/2 users.

In summary, because of the basic design features of the two operating systems, OS/2 is the ideal operating system for intermediate to power users in business and engineering fields who need sophisticated multitasking, memory protection, advanced interprocess communication, high performance or custom file systems, and advanced data-exchange capabilities. Typical OS/2 applications range from business applications (such as accounting, human resources, and relational database systems) to CAD (computer-aided design), desktop publishing, or complex "mission-critical" applications. In addition, these features make OS/2 the clear choice for network server based applications.

SYSTEM REQUIREMENTS

The hardware requirements for OS/2 and Windows are similar. A typical Windows workstation probably has a fast 80286 or 80386 with 2 MB of RAM, EGA or VGA graphics, a Microsoft-compatible mouse, and a 40 MB hard drive (of course, these figures will vary depending on the applications that a business needs to run). A typical OS/2 workstation probably has a fast 80286 or 80386 with 4 MB of RAM, EGA or VGA graphics, a Microsoft-compatible mouse, and a 40 MB hard drive or better. The primary distinguishing factor is that OS/2 typically requires about 2 MB of RAM more than a Windows workstation. Given today's prices for RAM, this is probably not a significant issue when deciding on which machine to purchase.

If system hardware costs are a big factor for your business's choice of operating systems, Windows has an edge (at the cost of less features). While the typical Windows workstation would be as described above, Windows can even run on an IBM PC-XT with an 8086 and 1 MB of memory -- though with greatly diminished performance.

FUTURE DIRECTION

Microsoft will continue to evolve both OS/2 and Windows in the future. The evolution of Windows will primarily focus upon user-oriented features. For example, Microsoft will continue to work on improving the Windows user interface, provide additional application integration, incorporate better font and color support, add multimedia interfaces, etc.

The evolution of OS/2 will parallel Windows evolution in the area of user-oriented features; that is, OS/2 will have the "same" user interface as Windows, the "same" application integration, the "same" font and color support, etc. As a matter of fact, Microsoft's goal is to make OS/2 versions 2.00 and later a superset of Windows. Microsoft is doing everything it can to allow as many Windows version 3.00 applications as possible to run unmodified under OS/2 version 2.00 itself, allowing current Windows users to carry over their Windows version 3.00 applications and run them directly under OS/2.

With the introduction of OS/2 version 2.00, OS/2 becomes a true 80386 operating system. It takes advantage of all the features available on an 80386 or 80486: demand paging, extremely large virtual memory, and a flat memory model allowing true 32 bit application support. It also requires a 80386 or 80486 and at least 4 MB of memory. If your application will be rolled out on this class of hardware, OS/2 will make the most of it, and is a good choice. In addition, OS/2 version 2.00 will provide comprehensive MS-DOS support with MVDMs (multiple virtual DOS machines), so a user won't have to give up anything when running OS/2.

Other features will be available over time. You can expect OS/2 versions 2.00 and later to support advanced capabilities such as workstation backup, MIS initiated installation of the operating system across a LAN, as well as providing hooks to allow MIS