

Microsoft Memo

To: bradsi philba ralph johnen karlst liascr bens rickett
From: Davidcol
Subject: Better DOS than OS/2: Draft analysis and proposal
Date: August 9, 1991

Please read this memo and supply me with your feedback ASAP. I will setup a meeting where we can discuss and make decisions once you get a chance to read this.

As a result of the latest Win 3.1 review with upper management, it was made clear that we needed to compete with OS/2's "Better DOS than DOS". It was decided at the review to improve display performance with this in mind. Fearing that just improving display speed for windowed DOS apps would not be enough to win on this front and thus we'd do work for no clear gain, I set off to do an analysis of what it would really take to compete with IBM on better DOS than DOS. My focus has been to understand where Windows is weaker than OS/2 in this area and recommend some features which allow us to compete. (In the interest of time, this memo does not contain the "complete" analysis of Windows vs OS/2 in the DOS app support area, I'll leave that up to marketing to pull together, the data is well known.)

Where are we weaker?

This is a list of all the areas where Windows 3.1 is weaker than OS/2 2.0 on supporting DOS apps.

1) Robustness.

OS/2 2.0 uses a DOS emulation kernel based on DOS 4.01. Much harder for DOS apps to bring down the system since MVDMS are self-contained boxes that can't corrupt the system or other applications. Windows shares a single copy of DOS across VMs. DOS apps can bring down the system if they trash DOS. New method in Win 3.1 can kill a bad DOS app safely as long as the app didn't trash DOS itself. The user is given a less severe message in this case.

2) Capacity.

By loading a "fake DOS" kernel into a VDM -- essential a DOS 4.01 MSDOS.SYS with the file system code replaced by traps to the OS/2 kernel -- Cruiser is able to present 620Kb of free DOS RAM. Even more could be made available if IBM used the DOS 5 HMA trick. Windows 3.1 using DOS 5.0 and loading device drivers and TSRs high, Windows can give DOS apps up to 625kb free, depending on how many UMBs are free.

3) Display performance and smoothness.

By implementing a high speed interface in the PM display drivers, windowed DOS app performance exceeds Windows by a noticeable amount. Operations likes DIRs and TYPEs are also faster from this optimization. In addition, DIRs and TYPEs are also much smoother in OS/2 since the window is updated much more often.

4) Disk performance and smoothness.

OS/2 has implemented SuperFAT. This promises to provide great disk performance by implementing lazy-write and read-ahead among other features. Smartdrive has neither of these and as a result is slower in some common scenarios. Another item that hurts Windows is the fact that while "in DOS", the system can basically do nothing else. (There are some

MS7026742
CONFIDENTIAL

exceptions to this when Fastdisk is installed.) This means doing things like DISKCOPY and FORMAT in Windows is slow and jerky. Disk intensive tasks like database searches and compiles can also appear slow and jerky.

5) Keyboard responsiveness

Entering text in a windowed DOS session "feels" slow and jerky compared to OS/2.

6) Usability Features

OS/2 has plenty of "features" not found in Windows.

- a) Windowed graphic DOS apps in all VGA resolutions including 320x200x256 color mode. Windows only supports standard 640x480 for windowed DOS apps.
- b) Variable font sizes for windowed DOS boxes. Windows does change the font in a windowed DOS box if the user changes the mode using the DOS mode command. 40 and 80 column modes are supported as is 43 line mode. A far cry from what OS/2 allows.
- c) Mouse control for windowed DOS app. In Windows, the mouse can only be used for selecting text in the windowed DOS app.
- d) PIFs are "built in" and accessible from the System menu and from the DOS apps icon in the Desktop manager. In Windows, they are separate files (which I think is good) but they are not accessible directly from the program manager's property dialog box.
- e) Title bar of DOS session dynamically changes to reflect currently running DOS app. Windows uses the icon in the program manager for a minimized DOS app, but it doesn't change dynamically.
- f) Spool DOS app print jobs to PM spooler.
- g) LIM 4.0 support in VMs.

What have we already done?

We've already done a number of things in Windows 3.1 to close the gap. These include:

- a) Fastdisk VxD. This provides a number of key benefits:
 - overlapped I/O between Virtual Machines. This will provide much more efficient multitasking between VMs as it allows a second VM some processor time while another VMs file system request is being done.
 - read-ahead and write-behind for the paging file.
 - run many more VMs than Win 3.0 did.
- b) More PIF files in the box so customers don't have to create or customize them.
- c) Graphical DOS apps (up to VGA 640x480) can run in a window as can background DOS applications.

MS7026743
CONFIDENTIAL

What can be done?

Since we are obviously really late in the Windows 3.1 product cycle, we should focus our efforts on doing the 20% of the total work to get 80% of the marketing benefit of doing everything. With this in mind, I've summarized the recommended work in the following table. None of these are new ideas.

Feature	Benefit	Cost
Update full lines of text	10% faster than current scrolling	Dev: 2 man days
Update on CR/LF for Int 10h using apps	Smooth scroll appearance for DIRs and TYPEs and others.	Dev: 2 man days
Minimum scroll rectangle code.	Move 51% fewer bits for DIRs. We are currently moving lots of blanks.	Dev: 2 man days
Use Cruiser display code for super-fast text out.	Increased performance for displaying DOS screen updates in VGA and SVGA drivers.	Dev: 15 man days
Update more quickly based on keyboard interrupts.	Faster response when typing in text.	Dev: 3 man days
Investigate ways to better floppy access performance and smoothness	Would give users "real" use of their floppy drive while in Windows. Would require a "fastflop" VxD.	unknown at this time.
Bambi disk cache.	Improved file operations for apps and file manipulations. Write operations greatly improved with write caching.	See Philba's spec and schedule.
User selectable fixed pitch fonts for windowed DOS sessions and DOS apps.	Let's the user tailor how they want the DOS app or DOS prompt to look and the screen size needed for full content display.	Dev: 3 more days
Scalable fonts when user sizes the windowed DOS app.	Let's user maintain full content view of the DOS app, but size windows to fit whatever size they want. Massive sex appeal here.	unknown at this time.
Mouse support for windowed DOS apps.	Let's users really run DOS apps in a window with full functionality and usability.	3-4 weeks. I'm still borderline on this feature.

MS7026744
CONFIDENTIAL