

Date: 8 May 1990 8:10 est
Via: vax
To: fking, fingari
Subject: Microsoft views on systems support.

This ramble gives you some idea of how others at MS think about their role as systems vendors.

You will note that Letwin's argument (reproduced below from comments to a usenet forum) about VCFPI can be paraphrased as:

we will not support VCFPI because 1-2-3 needs it, and we wouldn't want to break important applications such as 1-2-3.

Forwarded message

via: mailbox vax, dreed on VAX at Lotus (LDC) ** Message Received OK
From: BFRANKSTON "Bob Frankston"
Date: 7-MAY-1990 17:17:33.01
To: LDBVAX:dreed
Subject: FYI/Gordon Lettwin on Migration
Source-date: 7 May 1990 17:12 est

Forwarded message

via: mailbox news, newsxfer on VAX at Lotus (LDC) ** Message Received OK
From: UNIX@L: "news"
Date: 6-MAY-1990 18:18:55.18
Received: by DniMail (v2.0); Sun May 6 18:18:05 1990 EDT
From news Sun May 6 18:18:03 1990
Received: by lotus.com (4.1/SMI-4.1)
Id AA15348; Sun, 6 May 90 18:18:03 EDT
Path: lotus!unnet!microsoft!gordonl
From: microsoft!gordonl (Gordon LETWIN)
Newsgroups: comp.os.os2
Subject: Re: What Is Microsoft's Migration Strategy From DOS To OS/2? (LONG)
Summary: the realities of OS support
Message-Id: <54472@microsoft.UUCP>
Source-date: 5 May 90 20:00:58 GMT
References: <29439@cup.portal.com>
Organization: Microsoft Corp., Redmond WA
Lines: 190
Apparently-To: newsxfer@ldbvax.lotus.com

I want to thank Mr. Estes for a well reasoned constructive criticism. The OS/2 folks at Microsoft always appreciate hearing meaningful (or at least, rational :-)) comments and criticisms.

Mr. Estes addresses one of the most difficult and non-intuitive aspects of the software business, the customer migration path.

In article <29439@cup.portal.com>, Will@cup.portal.com (Will E Estes) writes:

> In my last post I made the argument that Microsoft is making it
> unnecessarily difficult to migrate to OS/2 by not supporting VCFPI.
>
> Gordon Letwin says:



002118
LOTUS

> < OS/2 could NEVER have protection if we supported VCPI...
 >
 > I agree with the premise (i.e., OS/2 cannot have protection with
 > VCPI) but not the conclusion (i.e., this means that OS/2 will
 > never be successful) nor the implied conclusion (i.e., that
 > supporting VCPI today under OS/2 means that you must 'always'
 > support it and hence that OS/2 will always be unprotected).
 >
 >
 > What is an acceptable migration strategy? 1) Support VCPI under
 > OS/2 and Windows. 2) Announce to the world that they have three
 > years to switch to DPMI, and at that time support for VCPI will be
 > withdrawn. When you tell people what you are going to do well
 > ahead of time, and if what you are doing is clearly in their best
 > interest, it is amazing how cooperative they will be. If, on the
 > other hand, what you are doing is forcing something that they do
 > not perceive to be in their best interest down their throats then
 > they will resist. Switching from VCPI to DPMI over a period of
 > time is clearly reasonable, and vendors would do it (and by the
 > time that three years had passed everyone would be using OS/2 and
 > so they wouldn't have much choice if they still wanted to have
 > their applications work with the operating system everyone was
 > using).
 >
 > What is an optimal migration strategy?
 > 2) Announce to the world that DOS will
 > no longer be sold as of 2001 (or some date well in the future). 3)
 > sell only OS/2 to OEMs, and sell it at the DOS price level.

First, let me say that all of this makes a lot of sense. This is the way
 that I used to look at things, this is the way that we did things when I
 worked as a systems programmer at a university computing center. This
 makes a lot of sense, but unfortunately the market doesn't make sense -
 at least, until you think it through.

For starters, we could say that we're only going to sell OS/2 to OEMs.
 The problem is, OEMs don't view themselves as agents for Microsoft where
 Microsoft is the center of the world and they're our handmaidens; they view
 themselves as the center of the world - they're making and selling computers -
 and 'we're' just a vendor to them, providing the operating system. Their
 focus isn't on a uniform software migration path, their focus is on
 selling machines. Regardless of the cost of the software,
 those making and selling 8086s can't use OS/2. Are these guys
 just going to voluntarily liquidate their companies? And what of the
 286/386 machines that customers want to buy with only 1 meg of memory?
 Instead, these guys will continue to ship DOS. The old DOS, if nothing
 else. If we terminate their license for that, then they'll ship some
 third party clone. Even if Mr. Estes had a computer company and he agreed
 with the reasons for terminating DOS, his competitors are going to still
 be shipping it, and he'll go out of business unless he ships DOS too.
 So the result of this was intended to force a migration to OS/2, but
 in fact it simply forced a migration away from Microsoft. And as for
 buying OS/2, after having DOS yanked the OEM is going to be reluctant
 to get dependent upon another Microsoft operating system else we go flakey
 on them again!

Forcing people away from Microsoft has two disadvantages. First, on a
 personal basis, it means that I'll have to go back to my old job on the
 fries line at McDonalds. And much worse, from the user point of view, is
 that there will no longer be a coherent standard; this will greatly slow down
 the development of PC software of all kinds.

002119
LOTUS

(Standards are obviously commercially valuable to the "owner" of the standard, but they're also valuable to the user. We're much better off with a standard OS of quality "5" than with 4 competing operating systems of quality "8", because of the critical mass of third party software. I recently got a microcontroller development kit from a chip maker; it interfaced via a RS232 port and it came with a floppy. The floppy didn't even say what operating system it was for, nor did the seller have to ask - it was, obviously, for THE operating system - DOS...) So our refusing to sell DOS would be damaging for the users as well as for us.

Now, lets look at the other part of the proposed strategy:

- > Support VCPI under
- > OS/2 and Windows. 2) Announce to the world that they have three
- > years to switch to DPMI, and at that time support for VCPI will be
- > withdrawn. When you tell people what you are going to do well
- > ahead of time, and if what you are doing is clearly in their best
- > interest, it is amazing how cooperative they will be.

Once again, how do we "withdraw" support for VCPI? You have all these users who paid big money for these VCPI apps, they're happy with them, and they're not going to junk them. These guys can't buy your new operating system that drops VCPI support. For example, I still run DOS on my electronics lab machines at home. I paid \$1000 for a video grabber board and it's driving software. That software only runs under DOS. I'm not going to buy a new board, and the maker isn't offering OS/2 versions. Likewise I have a universal PROM burner. It's driving software ALSO runs only under DOS. Substitute VCPI for DOS and you have the same effect. For some of my key applications I don't want to shell out \$400 each to upgrade. For a lot of other stuff, there *is* no upgrade offered.

After all these years, we can't even stop supporting FCBs! FCBs are the nastiest of interface structures, they prevent the OS from even being able to enumerate all the open files on the system! Yet programs still use them. I myself wouldn't buy a DOS that didn't support them. I've never written a program that used them, but do my frame grabber programs, or EPROM burner programs, or other third party apps use them? I don't even know.

OK, we've dropped one shoe. Because of the VCPI software out there - stuff that's at best expensive to replace, and often irreplaceable, say only 20% of users buy the new non-VCPI operating system. Now, how many ISVs are going to write their products to use the features of this new operating system? Very few, because very few want to do 100% of the work to produce a product that can be sold only to 20% of the machines. The other shoe is that even if we keep releasing new versions, the *standard* will remain frozen at the VCPI level. Very few non-bundled programs will use any new features because they need to remain compatible with all those other machines out there.

Will the day finally come when there are no more meaningful VCPI programs out there? Yes, it will. But it will be 10 years. It's nearly 10 years for FCBs, and I can assure you that we're not dropping them. An early popular version of DBASE uses them and we can't hose those users. This time interval takes even longer than the lifetime of a product because people will CONTINUE to write apps using the old, "obsolete" standard. Why not? It still works on all of the machines. *They've* got nothing to loose. If Microsoft calls them up and tells them not to do it, they'll tell us to go f*ck ourselves, it's THEIR product and THEIR schedule and we can go jump in the lake.

002120
LOTUS

> Switching from VCEI to DPMI over a period of
> time is clearly reasonable, and vendors would do it (and by the
> time that three years had passed everyone would be using OS/2 and
> so they wouldn't have much choice if they still wanted to have
> their applications work with the operating system everyone was
> using).

But the problem is, those apps are already out in the field at time T, and it's the OPERATING SYSTEM that's now incompatible. Say that we produce a new DOS that won't run Lotus 123. Inside every pack, we'll include a piece of paper that says,

"Lotus 123 doesn't run because they used a feature 4 years ago that we warned them we would drop. And so we did. So it's their fault, not ours."

I'm sure that the product will sell like hot cakes. People won't mind giving up their spread sheets, they'll say, "well, it's Lotus's fault, so that's OK. Heaven forbid that we penalize Microsoft because Lotus was short sighted. I'll just switch spreadsheet packages - time and money - or maybe just do them by hand. Or maybe become a McDonalds fries cook..."

The fact is, people DON'T do what is in their best interest, they do what yields short term gain. That's why people eat unsalty diets. That's why Reagan can please people by hyperexpanding the national debt to increase spending and lower taxes. That's why people litter. That's why so very few people save for retirement. The whole area is well known and has seen some study and experimentation, in psychology it's called "The Fallacy of the Commons."

Sure, sometimes it helps. I met with Lotus's top technical guys when OS/2 was being designed. They'd say that they wanted to do such-and-so and I'd tell them that I wanted them to do it "this way" rather than "that way" because it would fit in better with our future plans, and they'd usually say, "sure." They were smart guys, they could understand and handle the alternate approach, they felt some professional sympathy for our needs, so they were very helpful. But what if there wasn't a good alternative? What if "future compatibility" meant that they'd have to give up or cripple some function that they really liked? They probably wouldn't have done it.

And regardless, there are hundreds of ISVs out there who we DIDN'T get a chance to talk too, and who don't know enough to recognize trouble spots on their own, and even if we did talk to them, there's lots of them that would tell us to go get screwed; they've already written the pseudo code for the bad way and so long as the thing runs, they're going to ship it, and I should just go sit on a tack and twirl. I've had this exact response many many times in the past 5 years. I wish I could name names.

Well, this has gone way over length, so let me quickly close by saying that even though Mr. Estes' suggestions are rational and seem to make sense, and I in fact thought the same way 10 years ago, it's my experience - long, bloody, and consistent experience - that it doesn't work that way. Once you've offered a feature you can never, never take it away again. Or at least, not for 10 to 15 years.

Gordon Letwin
Microsoft

002121
121200
LOTUS