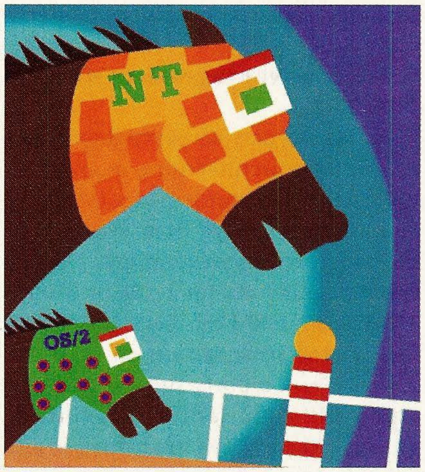
Windows NT

By Douglas A. Hamilton

Place Your Bets: 2.0 versus NT



SCOTT BALDWIN

OKAY, I GUESS I CAN stop beating around the bush. In the last couple of columns, my discussion of NT wasn't purely academic, something I'm sure many of you suspected. For the last couple of months I've been porting the Hamilton C shell to NT like crazy. As indeed, those of you who stopped by Microsoft's NT exhibit at Windows World/Comdex found me there, demonstrating on NT.

Any of you who have done booth duty at a major show (especially if you've had to do it single-handedly without relief) know several things:

1. The trick is to get a really big breakfast and skip the liquids.

2. Anyone with a roll of tape is going to be very popular when the show is over.

3. You'll hear the same questions over and over, and with all that practice, get very good at answering them.

Having now honed my answers to a fine edge, it seems a shame not to get a little more mileage out of them in a column.

Q. What's a nice OS/2 vendor like you doing in a place like this?

A. Microsoft invited me, IBM didn't. Lately, Microsoft has been making much more of an effort to court independent software vendors (ISVs). They seem to have more clearly figured out that if there are silly people out there willing to work long hours for not very much money developing applications for their systems, maybe it'd be in Microsoft's own interest not to discourage them. IBM could do as good a job, but right now they aren't.

Also—and this is really the more genuine reason—most of us ISVs are beginning to sense an opportunity in NT. I personally still think OS/2 2.0 will do well, but why pass up another market and another set of customers I couldn't reach with only an OS/2 product?

Q. How hard was it to port to NT?

A. Mechanical. My product is a text-windowed (kernel) application of about 85,000 lines of C, roughly half in utilities and the other half in the C shell command processor itself. Starting in February with the utilities, I spent about four to five weeks crunching source (lots of global search/replace with the editor, sticking in #ifdef's and so on,) including time to develop my own sense of how to translate from OS/2 to NT. After that, it was about a week to go from the first attempts to compile anything to having working utilities on a 386. The port to the MIPS was another day or two. The port of the C shell itself wasn't quite ready in time for Comdex—it feels like it needs another week or so—but has also gone very straightforwardly.

What this demonstrated to me is that the correspondence between OS/2 and NT functions is nearly one-for-one. In porting, there's very little debug because the algorithms hardly change. Also, the claim of portability to other machine architectures is real: If you have something that works on NT on an x86, getting it to work on the MIPS version is almost trivial.

Q. How stable is NT?

A. Very. Personally, it amazed me. The quality is very good, considering what I was working with was really alpha, not even beta, software. I found it to be roughly as bug-free and reliable as most any of the OS/2 2.0 betas.

Q. Which do you like better, NT or OS/2?

A. This isn't really a fair question yet, since OS/2 is clearly more mature overall and still has better application support. The current NT development kits are quite clearly just for developers building native NT software. The DOS boxes and the Windows 3.1 support aren't there yet, nor are many of the other "creature comforts" one might want. So I still do my day-to-day work in OS/2.

However, comparing the systems more in the abstract and considering how each is architected, it's more difficult to have a preference. They're both good systems.

Q. How big is NT compared to 2.0?

A. Despite what you may have heard, about the same. Both NT and 2.0 should require about 30 to 40 megabytes (MB) for a full installation, plus swap space. I doubt anyone will be choosing one system over the other based on size. There just isn't enough difference.

Q. How's the performance?

A. Again, despite what you may have heard, not bad. On my 16MHz 386, it is sluggish but it's certainly not off by an order of magnitude. I'd judge it to be off by maybe 2x to 3x, meaning that even with a 33MHz 386, I'd expect performance to be reasonably acceptable. Also, performance isn't uniform: The display painting is relatively slower than the rest of the system. The bad news is that it's the display speed that everyone sees first when they try the system. The good news is that as a localized problem, and particularly, one that isn't at the core of architectural issues like how the system gets layered around the microkernel, it's a lot easier to fix. One Microsoft engineer advised me the problem is actually very specific: a poor bitblit routine.

My sense is that while NT may or may not be quite as fast as 2.0 by the time it's released (I honestly don't know how to call this yet), they'll be sufficiently competitive that this also may not be the factor in purchase decisions.

Q. What advantages does NT have over 2.0?

A. The big ones are portability to other hardware architectures, support for symmetric multiprocessors (SMPs), a more natural lineage from Window 3.x and better security mechanisms. NT is already running on Intel x86 and MIPS R4000 RISC, and I can personally vouch that it's the same system on both machines. Press reports suggest the DEC Alpha will soon follow. In addition, sources tell me Hewlett-Packard is also considering NT for their Precision Architecture.

Running on an SMP, NT is capable of distributing the load across all available processors, transparent to the applications. Aside from writing it as a multithreaded application, the designer need not do anything special to use SMPs under NT. So although SMPs are not yet a significant factor in the micro industry, the ready support NT offers for them could change that quickly, benefiting companies like NCR and others who have already made big investments in SMP design.

Q. What advantages does 2.0 have over NT?

A. Spell the big one IBM. IBM still has a tighter hold on the corporate MIS market.

You'll see that commitment to OS/2 rewarded with better host and large network connectivity, technical support and bug fixes from IBM and other features tailored to that market.

Also, speaking purely technically, I do expect the OS/2 2.0 Virtual DOS Machine (VDM) capability to outstrip that provided in NT, though by how much is uncertain. Also, there's a very real possibility, despite complaints from longtime OS/2 users, that you could indeed judge 2.0's new Workplace Shell as a superior, easier-to-use interface than Windows.

Q. Can IBM really make any money on 2.0?

A. Not at $49 (the price for an upgrade from Windows). With 21 high-density disks (and more to come in planned Corrective Service Diskette releases—IBMese for bug fixes), a royalty to Microsoft rumored to be around $20, documentation and so on, IBM cannot possibly be making money. Don't expect that price to last.

Q. When will NT be real?

A. Microsoft claims they'll be moving into a massive beta this summer—thousands of copies, available almost on demand—and full product release around the end of the year. Looking at the quality of the current alpha, I see nothing to cause me to question those claims.

Q. Which do I think's going to win?

A. Both NT and OS/2 2.0 are very good systems, very competently designed with core functionality sets that are very similar. Proponents of either will find reasons to prefer their own choice. I don't think either system has a technical "knock-out" capability against the other. They're just too evenly matched.

Instead, I believe it'll come down to a question of whose marketing is more effective and with whom. IBM will, I expect, continue to do slightly better with large corporate users. OS/2 appears to have roughly an eight-month lead, and estimates that 2.0 will sell 2 to 4 million copies this year seem entirely plausible. However, among the larger population, particularly individual buyers, I think Microsoft holds the edge.

The smart money is going to be on both systems doing pretty well. ■

Douglas A. Hamilton is president of Hamilton Laboratories in Wayland, Mass., and author of the Hamilton C Shell, an advanced interactive command processor and tools package for OS/2. Reach him on BIX as hamilton or care of Editor at the address on page 10.