

Better Windows?

BY DOUGLAS A. HAMILTON

NT is Powerful, Real and Just Around the Corner

SURPRISE! NT is a lot more real than most of us ever expected. A few months ago, many of us were ready to write off NT as just a lot of smoke intended to disrupt IBM's OS/2 2.0 efforts. Few had even seen NT. There were reports of perhaps a half-dozen early, early betas given out at selected sites around the country. Reports leaking out of those sites were decidedly mixed. The whole thing was cloaked in stifling secrecy. It was impossible to find out anything of substance. How could anyone believe Microsoft would ever be able to release NT on the schedule they'd been talking about? End of 1992? Please!

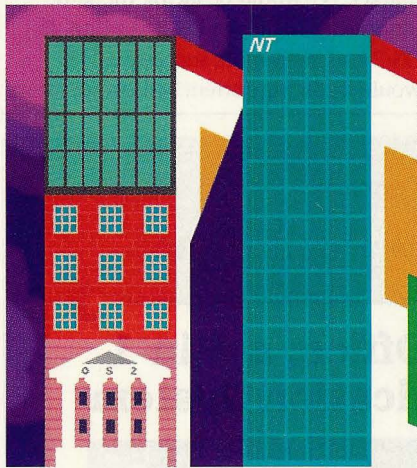
Now it looks like they weren't kidding. Microsoft has shown NT running on both Intel and MIPS RISC machines at public shows. The complete specifications for the programming interfaces have been published and are readily available. Microsoft is beginning to recruit selected ISV's (independent software vendors)

with invitations to technical briefings and offers of immediate delivery of the complete NT SDK (software development kit) on CD-ROM. It's real, even if still not a product.

Developers often talk about the "tired code" syndrome, the fact that old code passed through many hands gets fragile. So many of the underlying assumptions and design considerations having been lost or forgotten, so no one really knows very well how the whole thing works. Writing new code, particularly in a high level language, is just plain faster. It's common knowledge in developer circles that much of the OS/2 code is ancient assembly, but that NT was written from scratch in C. I think we've fallen victim to considering only the lead OS/2 2.0 started with, rather than likely rate of progress. My guess is we'll see both OS/2 2.0 and NT reaching peak stride within six months of each other—maybe less.

Yes, there are those who decry the fact that it takes a whole CD-ROM to hold NT. But that one CD holds both MIPS RISC and Intel x86 versions along with all the tools. The actual machine requirement for an end-user isn't much different than for OS/2 2.0. In contrast to IBM's strategy of placing OS/2 on every desktop (meaning it must be all things to all people), Microsoft seems to be pursuing a segmentation strategy: Windows and DOS on the low end, NT on the high end. They also seem to be positioning NT much more clearly as an alternative to UNIX, something IBM has never done with OS/2. These factors should give Microsoft a little more insulation from claims that NT is too big or complex; after all, you get what you pay for.

NT does hold advantages over 2.0—



Better Windows?

portability to RISC processors, support for symmetric multi-processing, memory-mapped files, C2 security, block-structured exception handling, better support of debuggers and, overall, more of an object-oriented approach, all of which may translate into better applications for the end-user.

But don't just take my word for it. In an unprecedented move, Microsoft allowed their white paper describing the strategy behind NT, and even the complete specifications for the Win32 API (Application Programming Interface), to be posted on public bulletin boards. On BIX, for example, these materials are available in the `ibm.windows/listings` area as `ntwin32.lzh` (the white paper) and `api32w-1.zip` through `api32w-4.zip` (the API specification in Windows Help format). Be forewarned: This is not just some fluffy overview; it's the complete meat and potatoes and a lot to digest. Even compressed, these files total more than 2 megabytes.

What you'll see is that NT is not the

disjoint break from the past that OS/2 was at its introduction. OS/2's threads, processes and semaphores left most DOS developers completely lost. Unless they had some background on more sophisticated systems like UNIX, the pragmatic implications of having to coordinate lots of asynchronous activities were extremely difficult to master and a good reason why it took so long before we saw applications that could take advantage of OS/2. And if highly trained developers had trouble understanding OS/2, imagine the plight of the end-user!

As a much more evolutionary refinement, building on the OS/2 experience, NT should have a much easier acceptance. In a very real and ironic sense, OS/2 has been just exactly successful enough to give NT its best possible opportunity. If OS/2 had been a complete (well, more complete) disaster in the market, NT would be facing the same pioneering challenge OS/2 ran into. If OS/2 had been more successful, NT would be facing entrenched competition.

Windows developers will have an almost trivial job moving to NT. Even if they do nothing, their applications will still run on NT since it is promised to be compatible with all existing Windows binaries. (On the MIPS platforms, NT will emulate the x86 architecture in software for those applications that require it.) Even a full-blown port to NT should be little more than some editing and a recompile.

Nor will the job of porting to NT be that difficult for OS/2 developers. The graphics interface is certainly different—it's Windows, not PM—but anyone who's been writing to the OS/2 kernel API is going to feel right at home. The similarities to OS/2 are immediately apparent. Sure, all the names have changed, but mostly all the same functions are there and they operate in mostly all the same ways.

Differences with OS/2 demonstrate intriguing refinements. An example is the consistent use of the handle notion. Everything has handles, not just files. There are handles for processes, threads and everything else. To wait for a completion, one need only wait on the handle. All I/O is naturally asynchronous; to wait for the completion, simply wait on the handle. To wait for a child process to exit, again, just wait on the handle. This is a nice simplification of very disjointed style in OS/2, where everything was a special case. For example, waiting for a child process required different methods depending on whether the child was running in the same or a different window.

My suspicion is that vendors who have focused solely on OS/2 (presumably because their applications required the advanced facilities that only OS/2, rather than DOS, could offer), might do well to begin looking more closely at NT. Something tells me there could be a market for them later this year. ■

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. He can be reached on BIX as hamilton, on MCI Mail at 389-0321, or by letter care of editor at the address on page 8.

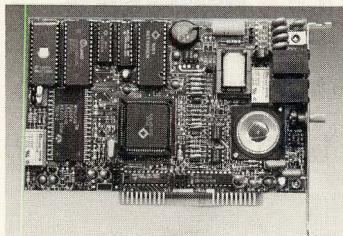
Complete Communications Solutions...

...like the Home Office, a 4-in-1 Windows Communication System

The Home Office adds a voice feature to a combination data/fax modem to turn your PC into an answering machine! Also functions as a line switcher with its ability to distinguish between data, fax or voice calls.

MAJOR FEATURES:

- Data modem with
- V.42bis and MNP-5 data compression/error correction
- 9600 bps send/receive fax
- Voice capability
- Communications software
- Windows fax/voice software



The Home Office - available in two versions, one with a 2400 bps data modem, the other with a 14,400 bps (V.32bis) modem. Both are internals, with V.42bis/MNP-5, and identical fax and voice features.

Retail price:

Home Office (2400 bps version).....\$299.
Ultima Home Office (14,400 bps).....\$499.



(800) 477-3473 • (503) 692-9600 • Fax (503) 691-1101

Circle Reader Service No. 168