



## Unix or NT? Why, NT, of course

**T**WICE IN THIS ISSUE of *EDN*, I get to address the subject of EDA workstations—here and in the article on pg 90. The article focuses on workstation performance and examines both Windows NT and Unix systems in single- and dual-processor configurations. In the article,

I advocate comparing systems based on the performance of each system. For the most part, the hardware dictates performance. Still, I know there's an additional intangible software angle to any such comparison and in some cases a clear-cut technical reason to choose a system based on the OS. It's clear, however, that some of you in the engineering community prefer a Unix system, no matter how fast the underlying hardware performs and no matter how much it costs. I'd like to take the other side here and explain why many, if not most, of you should prefer NT.

Let's start with the short answer. From an end user's perspective, Windows NT is far superior to Unix. And most of you are end users when it comes to EDA tools and workstations. I realize that some of you are software developers, and, if you are developing software for Unix or open systems in general, then you need a Unix-development host. And, before we go further, back off Mac fans. I know that your OS is even easier to use and hardware support is even more seamless. But the Mac is a closed system, and that fact limits support from software and hardware vendors. The lack of EDA tools on the

Mac eliminates it from comparison with Unix or Windows NT here.

Back to Windows NT's end-user superiority. You can't argue with this point. Just compare trying to install a modem or a printer on a Unix system with doing so on an NT system. Windows NT is light-years ahead in establishing a standard abstraction layer for a wide and always-expanding hardware base. For most popular peripherals—from printers to modems to cameras to scanners—you don't even need a driver from the peripheral manufacturer. Windows NT 4, with the latest service pack, includes the drivers. It's even far easier to get a Windows NT LAN configured than it is a Unix LAN—despite the fact that built-in networking has long been heralded as a feature that differentiates workstation-class systems and OSs from the PC world.

I could go on for days. All Windows software comes with consistent installation routines and follows well-established conventions about where to install files, including shared system files, such as multimedia libraries. The Windows Control Panel provides a centralized and intuitive facility that you use to make system-level hardware and software changes. To accomplish the same thing on a Unix system, you usually need to locate settings buried in a configuration file and manually edit the file using a text editor. Windows Explorer is easier to use and configure than the File Manager program in the Unix Common Desktop Environment (CDE) or any other Unix file manager I've tried. At first, I couldn't even figure

out how to rename a file using the CDE File Manager. Surely the program's designers could have added a rename command to the menus. And, despite the fact that you can carefully click and edit file names directly, the help system doesn't point you in that direction. When I asked a friend about renaming files, he said, "Just go to Terminal and do it from the command line."

This experience indicates what I presume is the Unix mentality. Real computer users don't need niceties; they know how to go to the command line and accomplish anything via brute force. I remember thinking in a similar manner when I first tried Windows 2.0. The software was slower than slow, and it didn't accomplish anything that I couldn't already do in DOS. I became a Windows user only because the AmiPro ([www.lotus.com](http://www.lotus.com)) word processor included a table generator that was invaluable for preparing the product-comparison tables I regularly used in *EDN* articles. It was only later that I realized that Windows' greatest gift is hardware abstraction rather than a fancy graphical user interface (GUI).

I know some of you are screaming now—telling me about Unix application-installation utilities, telling me about how you can install icons for applications into one of several program-launch windows, telling me that your system has a tool to manage printers or modems. I tried all that stuff. You can make it work, but it's archaic compared with Windows NT.

I honestly approached this evaluation project excited about the prospect of using Unix and discovering what I've been missing. I hadn't used Unix much, because the command line is virtually the only way to accomplish anything. Yet, I always hear about how Unix is more robust and almost a requisite for real computer users. I have to say that I was shocked by the state of Unix. The CDE GUI looks nice. It's easy enough to move and resize windows. But it's an empty GUI front end with none of the middleware meat backing it up.

To make matters worse, no two Unix systems are alike. And I'm not



talking Sun versus Hewlett-Packard versus Digital versus IBM Unix (although that is another problem). The fact is that two Solaris installations can be far different based on who does the installation. Let me explain: In installing the applications I used in my workstation tests, including a software-based license manager, I had to follow pages of installation instructions. I had to set environment variables and paths—no real challenge but a strange thing for a Windows NT user to do manually. You have to place the definitions in a start-up file of your choice. You could choose the *.profile* file that runs when a user logs on, or you could choose a system start-up file so that the definitions are present regardless of who logs on. Well, the instructions list about a half-dozen system files that might make a good home to the variable definitions—none of which are on my system. There are similarly named files but none that exactly match the instructions.

For an even worse nightmare, try installing a resource, such as a software license manager. The license manager works via a daemon that you must kick off at some point. Again, you can start the daemon from a user log-on or at boot time, and you choose the start-up file in which the commands go. But correctly crafting the commands that start the daemon is no slam dunk. The instructions list a number of similar Unix commands that you choose based on which shell is active. Well, the active shell probably depends on which shell is the default choice for the current user, or maybe not, because a simple command somewhere else in the start-up sequence could have changed shells. The active shell is likely different for a command that executes at system start-up from that for a command that executes when a user logs on. And, in addition to matching the command to the shell, the command arguments depend on which directory you decide to place the license file in and other variables. I knew I was in trouble when the instructions concluded with a sample script file. What I think the author of the manual intends as a helpful hint sounds more like a disclaimer. It basically warns

that one type of HP Unix installation had successfully used the sample script. What's up with that?

Are you telling me that the supposedly brightest OS minds in the business can't adopt a more uniform directory structure? Just choose a shell—any shell. You don't need a half-dozen shells in which 75% of the commands are the same and 25% couldn't be more different. Further, Unix implementations from different companies are vastly different.

I guess if you are a Unix user and have a full-time system administrator that installs applications and peripherals for you, then you can get on to the task of using the system. After all, the computer should still be a tool that you use to increase your productivity, whether the application is an

## ARE YOU TELLING ME THAT THE SUPPOSEDLY BRIGHTEST OS MINDS IN THE BUSINESS CAN'T ADOPT A MORE UNIFORM DIRECTORY STRUCTURE?

EDA-synthesis tool or a word processor. If you don't have a system administrator, you'd better have a better understanding of Unix than any end user should expect to need.

I suspect that you Unix zealots are now screaming about flexibility and configurability. I just don't see what these features buy. Why can't the same file with the same name perform the same start-up duties on every system? I realize that Unix is inherently a multiuser system. You need some flexibility to set the environment for each user. But even system administrators don't need the flexibility available in packages such as Solaris.

So, let's address the only real reason an end user might choose Unix: Presumably, Unix is more reliable than NT, and I can't challenge that claim. I've never had an NT system crash, so I suspect that the hype is louder than the reality. I know Windows NT gets more reliable with each release and service pack. Moreover, I know designers that use nothing but NT running on no-name clone systems and are successfully designing ICs. I find that the acceptance of NT for such tasks is widespread among designers at start-ups and small companies in which the work force has an equity

interest in the bottom line. Moreover, as related in my workstation article, Mounir Hahad of Silvaco, who prefers Unix, also reports running a six-day SmartSpice simulation on an NT machine without a problem.

The ability to develop extensive script files under Unix also enamors some designers. Out of the box, NT basically offers DOS batch-file capability, although third-party utilities greatly improve scripting capabilities. But you can run Unix scripts on an NT machine. I'd prefer a GUI in an application that allows me to schedule sequential operations. But I also tested the Cygnus Gnu-Win32 utilities, which are free at [www.cygnus.com](http://www.cygnus.com) and include a Unixlike shell that runs on NT. You can seamlessly run your Unix scripts under the shell.

I can come up with but two reasons an EDA end user would choose Unix. First, the most powerful hardware platforms run only Unix. The Intel-based systems surpass low-end Unix workstations in performance and equal Unix midrange performance. Still, the fastest systems from HP, Sun, and others run only Unix—except for those Compaq Alpha screamers that might just be the industry's fastest and run a choice of Windows or NT. Second, some high-end applications require a 64-bit address space, and only Unix provides that support.

I just don't see an obstacle to NT's dominating the engineering desktop. The NT hardware offers more bang for the buck by a long shot. Unix-application servers can sit on the LAN in the background, and users can farm out tasks when processing needs exceed the capabilities of the desktop. Moreover, I know all you Unix zealots use Windows 9X or NT for office-productivity tools anyway. Swap to NT, and you can have everything on one system. One final note: Everything I say about Unix also applies to Linux. I don't need another OS that I have to administer. Microsoft isn't perfect, but Windows NT is a damned good OS when you give it a chance.