If Windows 3.11 required a 32-bit processor, why was it called a 16-bit operating system?

devblogs.microsoft.com/oldnewthing/20100517-00

May 17, 2010

Raymond Chen

Commenter Really16 asks via the Suggestion Box how 32-bit Win32s was, and why Windows 3.11 was called 16-bit Windows when it required a 32-bit CPU and ran in 32-bit protected mode. First, let's look at how Windows worked in so-called Standard mode. Actually, it was quite simple: In Standard mode, Windows consisted of a 16-bit protected-mode kernel which ran applications in 16-bit protected mode. I suspect there would be no controversy over calling this a 16-bit operating system. With the introduction of Enhanced mode, things got more complicated. With Enhanced mode, there were actually three operating systems running at the same time. The operating system in charge of the show was the 32-bit virtual machine manager which ran in 32-bit protected mode. As you might suspect from its name, the virtual machine manager created virtual machines. Inside the first virtual machine ran... a copy of Standard mode Windows. (This is not actually true, but the differences are not important here. Don't make me bring back the Nitpicker's Corner.) The other virtual machines each ran a copy of MS-DOS and were responsible for your MS-DOS sessions. Recall that Enhanced mode Windows allowed you to run multiple MS-DOS prompts that were pre-emptively multi-tasked. These other virtual machines ran in a variety of modes, but spent most of their time in virtual-86 mode. MS-DOS applications could use the DPMI interface to switch into 16-bit protected mode, or even 32-bit protected mode if they wanted to. (And that's how Standard mode Windows ran inside the first virtual machine: It used the DPMI interface to switch to 16-bit protected mode.) It's kind of stunning to realize that Enhanced mode Windows was really a completely new operating system with multiple virtual machines, pre-emptively multi-tasked with virtual memory. In principle, it could have created a virtual machine and hosted yet another random operating system inside it, but in practice the only two operating systems it bothered to host were Standard mode Windows and MS-DOS. Enhanced mode Windows was called a 16-bit operating system because it ran 16-bit Windows applications (inside a "Windows box", you might say). The supervisor operating system was a 32-bit operating system, but since applications didn't run in supervisor mode, that really didn't mean much. For all anybody cared, the supervisor operating system could have been written in 6502 assembly language. As long as it does its supervisory job, it doesn't matter what it's written in. What people care about is the applications that you could run, and since Enhanced mode Windows ran 16-bit Windows

applications, and since it ran a copy of 16-bit Standard mode Windows to do all the things that people considered Windows-y, it was the number 16 that was important. Besides, you can imagine the uproar from the Slashdot crowd (if Slashdot had existed back then) that an operating system whose purpose is to run 16-bit applications in a 16-bit GUI environment would dare call itself a 32-bit operating system. How 32-bit was Win32s? Pretty darned 32-bit. A Win32s application ran in the same virtual machine as the rest of Standard mode Windows, but when it ran, the CPU really was in 32-bit protected mode. Naturally, it did all its work under the supervision of the virtual machine manager, and it had to coordinate its work with the Standard mode Windows kernel that it was sharing the virtual machine with. But when your 32-bit application was running, you were bought in: Your registers were 32-bit, your pointers were 32-bit, you accessed data in a 32-bit data segment, and you executed 32-bit instructions out of a 32-bit code segment.

[The blog server is undergoing a long-overdue upgrade which will take a week to complete. During that time, comments will be disabled.]

Raymond Chen

Follow