Why did disabling interrupts cause Windows 95 to hang?

devblogs.microsoft.com/oldnewthing/20151204-00

December 4, 2015

Raymond Chen

One of the ways people used to demonstrate the awfulness of Windows 95 was that you could hang the system with this simple two-instruction MS-DOS program:

```
cli
jmp $
```

The `cli` instruction disabled interrupts, and the `jmp $` instruction merely jumped to itself. This spun the CPU with interrupts disabled, hanging the system. There is no way to break out of the loop because you told the CPU to stop listening to the outside world.

Why did Windows 95 let MS-DOS applications disable interrupts?

Compatibility, of course.

In principle, Windows 95 (and Windows 3.1) could have virtualized the interrupt flag for MS-DOS programs. If a virtual machine disabled interrupts, it would disable interrupts only for itself; other virtual machines would still have interrupts enabled, and interrupts would still be enabled in the virtual machine manager. Indeed, if the program was running in protected mode, the interrupt flag *was* virtualized. There is a special case for code running in virtual 8086 mode. Why the special exemption just for v86-mode?

There were a lot of MS-DOS drivers which relied on timing loops and tight polling. If you virtualized the interrupt, then the virtual machine that disabled interrupts would have a messed-up timing loop because its loop would be interrupted by other virtual machines that were also running at the same time. Similarly, the tight polling loop could miss an event because the hardware gave you only a 10ms window to respond to the signal, but the virtual machine got pre-empted for 55ms due to multi-tasking. That would cause your scanner to return garbage data, or your tape backup to fail, or your CD-ROM burning software to create a coaster.

Windows 3.1 (and Windows 95) addressed this problem by disabling multi-tasking when a virtual machine disabled interrupts. Disabling interrupts allowed a virtual machine to prevent other virtual machines from stealing CPU and messing up its hardware timing and

polling loops.

It was the general impression that end-users would prefer to use the hardware that they paid good money for, and which was working just fine in MS-DOS. (Back in these days, a low-end CD-ROM drive cost around $200. I owned one such, and the only driver it came with was an MS-DOS driver.)

Of course, Windows NT addresses this problem a different way: It simply doesn't support MS-DOS drivers. But in the early 1990's, a lot of hardware devices didn't have drivers for Windows NT (and a lot of computers didn't meet Windows NT's hardware requirements), so your choices were limited.

- Stick to MS-DOS and don't upgrade.
- Suck it up and run Windows 95.
- Use your external CD-ROM/Bernoulli/ZIP/tape drive as a doorstop.

Raymond Chen

Follow