The world's slowest RET instruction

devblogs.microsoft.com/oldnewthing/20060103-01

January 3, 2006

Raymond Chen

Occasionally, somebody will ask

> I'm debugging a hang, and I see that many threads are stuck at a RET instruction. When I try to trace one instruction from that thread, the trace breakpoint never fires. It's as if the RET instruction itself is wedged! I've found the world's slowest RET instruction.

(A common variation on this theme is that the thread in question is consuming 100% CPU... on a RET instruction?) Because what you see in that RET instruction is a thread that is executing in kernel mode. The kernel parked the user-mode side of the thread at a RET instruction, poised to execute once the kernel-mode side has returned. Which it hasn't yet. In order to see what is really going on with that thread, you have to drop into the kernel debugger. You might be able to make some educated guesses (also known as "invoke psychic powers") based on what you can still see on the user-mode side. For example, the RET could be returning back to a `WaitForSingleObject` call, which tells you that whatever this thread is waiting for hasn't happened yet.

[While Raymond was on vacation, the autopilot stopped working due to a power outage. This entry has been backdated.]

Raymond Chen

Follow