

The quiet dream of placebo settings

 devblogs.microsoft.com/oldnewthing/20061122-50

November 22, 2006



Raymond Chen

Back in the Windows 95 days, people swore that increasing the value of `MaxBPs` in the `system.ini` file fixed application errors. People usually made up some pseudo-scientific explanation for why this fixed crashes. These explanations were complete rot. These breakpoints had nothing to do with Windows applications. They were used by 32-bit device drivers to communicate with code in MS-DOS boxes, typically the 16-bit driver they are trying to take over from or are otherwise coordinating their activities with. A bunch of these are allocated at system startup when drivers settle themselves in, and on occasion, a driver might patch a breakpoint temporarily into DOS memory, removing it when the breakpoint is hit (or when the breakpoint is no longer needed). Increasing this value had no effect on Windows application. I fantasized about adding a “Performance” page to Tweak UI with an option to increase the number of “PlaceBOs”. I would make up some nonsense text about this setting controlling how high in memory the system should place its “breakpoint opcodes”. Placing them higher will free up memory for other purposes and reduce the frequency of “Out of memory” errors. Or something like that. I was reminded of this story by my pals in products support who were trying to come up with a polite way of explaining to their customer that there is no /7GB boot.ini switch. In other situations, they sometimes dream of shipping `placebo.dll` to a customer to solve their problem.

(And by the way, the technical reason why the user-mode address space is limited to eight terabytes was given by commenter darwou: The absence of a 16-byte atomic compare-and-exchange instruction means that bits need to be sacrificed to encode the sequence number which avoids the ABA problem.)

Raymond Chen

Follow

