There is no /8TB flag on 64-bit Windows

devblogs.microsoft.com/oldnewthing/20150605-00

June 5, 2015

Raymond Chen

A customer reported that their 64-bit application was crashing on Windows 8.1. They traced the problem back to the fact that the user-mode address space for 64-bit applications on Windows 8.1 is 128TB, whereas it was only 8TB on earlier versions of Windows for x64. They wanted to know whether there was a compatibility mode that they could set for their application to say “Give me only 8TB please.”

No, there is no compatibility mode to shrink the user-mode address space down to 8TB.[1] You get it all, like it or not.

Allowing 64-bit applications to opt out of the 128TB address space has implications for the rest of the system, such as reducing the strength of ASLR.

As for how they ended up having a dependency on the address space being at most 8TB, they didn’t say, but I have a guess: They are using the unused bits for tagging.

If you are going to use tagged pointers, you need to put your tag bits in the *least* significant bits, since those are bits you control. For example, if you align all your objects on 16-byte boundaries, then you have four available bits for tagging. If you’re going to use upper bits for tagging, at least verify that those upper bits are available.

[1] If anything, the user-mode address space will only grow over time. The original 8TB limit in earlier versions of Windows was due to the lack of a CMPXCHG16B instruction, but now that the instruction is available (and support for it deemed mandatory), the full 256TB address space is available to the operating system, and right now, it decides to split the address space evenly between user mode and kernel mode.

Raymond Chen

Follow