It’s the address space, stupid

devblogs.microsoft.com/oldnewthing/20130628-00

June 28, 2013

Raymond Chen

Nowadays, computers have so much memory that running out of RAM is rarely the cause for an “out of memory” error.

Actually, let’s try that again. For over a decade, hard drive have been so large (and cheap) that running out of swap space is rarely the cause for an “out of memory” error.

In user-mode, the term *memory* refers to virtual memory, not physical RAM chips. The amount of physical RAM doesn’t affect how much memory a user-mode application can allocate; it’s all about commit and swap space.[1] But swap space is disk space, and that is in the hundreds of gigabytes for hard drives. (Significantly less for SSDs, but even in that case, it’s far more than 4GB.)

The limiting factor these days is address space.

Each thread’s stack takes a megabyte, and if you’re creating a lot of threads, that can add up to a lot of address space consumed just for stacks. And then you have to include the address space for the DLLs you’ve loaded (which quickly adds up). And then there’s the address space for all the memory you allocated. (Even if you don’t end up using it, it still occupies address space until you free it.)

Typically, when you get an `ERROR_OUT_OF_MEMORY` error, the problem isn’t physical memory or virtual memory. It’s address space.

This is one of the main benefits of moving to 64-bit computing. It’s not that you actually are going to use or need all that memory. But it relieves pressure on the address space: The user-mode address space in 64-bit Windows is eight *terabytes*.

When the day comes that eight terabytes is not enough, we at least won’t have to redesign the application model to expand the address space. The current x86-64 hardware has support for address spaces of up to 256TB, and the theoretical address space for a 64-bit processor is sixteen exabytes.

[1] Of course, physical RAM is a factor if the application is explicitly allocating physical memory, but that's the exception rather than the rule.

Exercise: Help this customer clear up their confusion: They reported that processes were failing to start with `STATUS_DLL_INIT_FAILED` (0xC0000142), and our diagnosis was that the desktop heap was exhausted. "The system has 8GB of RAM installed, and Task Manager reports that only 2GB of it is being used, so it is unlikely that I am running out of any kind of heap/memory."

Raymond Chen

Follow