Windows 95 doesn’t boot with more than 1GB of RAM

devblogs.microsoft.com/oldnewthing/20030814-00

August 14, 2003

Raymond Chen

Short version

Windows 95 will fail to boot if you have more than around 480MB of memory. (This was considered an insane amount of memory back then. Remember, Windows 95’s target machine was a 4MB 386SX and a powerful machine had 16MB. So according to Moore’s law, that gave us seven years before we had to do something about it. One of my friends got 96MB of memory on his machine to test that we didn’t tank under “insanely huge memory configurations” and we all drooled.)

Windows 98 bumped the limit to 1GB because there existed a vendor (who shall remain nameless) who was insane enough to want to sell machines with 1GB of RAM and preinstall Windows 98 instead of the much more suitable Windows NT.

Long version

One of the first things that happens in the Windows 95 boot process once you have transitioned into 32-bit mode is to initialize the 32-bit memory manager. But now you have a chicken-and-egg problem: The memory manager needs to allocate some memory in order to keep track of the memory it is managing. (Keeping track of which pages are paged in and which are paged out, that sort of thing.) But it can’t allocate memory until the memory manager is initialized. Eek!

The solution is to initialize the memory manager twice.

The first time the memory manager is initialized, it gets all its memory from a fixed block of memory preallocated in the init-data segment. It sets up this fixed block as the memory manager heap. So now there is a heap available to satisfy memory allocations.

Next, the memory manager starts looking for the *real* memory in the system, and when it finds some, it allocates memory (from the initial fixed block) to keep track of the real memory.

After the memory manager has found all the real memory in the system, it’s time to initialize the memory manager a second time: It carves out a chunk of that real memory to use as the “real heap” and copies the information from the heap that it has been using so far (the fixed-sized heap) to the “real heap”.

Once everything has been copied and all the pointers fixed up, the global memory manager heap pointers are changed to point at the new (“real”) heap and the original heap is abandoned.

The memory consumed by the original heap is reclaimed when the init-data segment is discarded (which happens at the end of system initialization).

The total RAM limitation occurs because the size of the fixed block in the init-data segment needs to be large enough to satisfy all the memory allocations performed during the memory scan. If you have too much memory, an allocation during the memory scan fails and the system halts.

The size of the init-data segment was chosen to balance two factors. The larger you make it, the more memory you can have in the system before hitting an allocation failure during the memory scan. But you can’t make it too large or machines with small amounts of memory won’t even be able to load VMM into memory.

Raymond Chen

Follow