Getting MS-DOS games to run on Windows 95: Too much EMM memory

devblogs.microsoft.com/oldnewthing/20160425-00

April 25, 2016

Raymond Chen

A report arrived on an MS-DOS game that crashed when run on Windows 95. Mind you, the game ran for quite a while before finally keeling over. The repro steps were “Start the game, then go to City 1, then City A, then City Alpha, then City , then City . When you go to City , the game crashes.”

I gotta say: Our testers were quite thorough.

The reason for the failure is that the program saw too much EMM memory.

Well, not exactly.

Each time you move to a new city, the program tries to allocate some more EMM memory. Under normal conditions, it runs out of EMM memory at some point and presumably starts swapping data out to disk.

What happens when run under Windows 95 is that when you move to the fifth city, the program makes its 65th request for EMM memory. And under Windows 95, the call succeeds, because Windows 95 tries to be all awesome like that. The program then saves that EMM memory handle into a table. The table is a fixed size.

That fixed size is 64 EMM handles.

The program worked when running under MS-DOS because the MS-DOS EMM386 driver defaults to 64 EMM handles, so when the program makes its 65th request for EMM memory, it gets the error “no more EMM handles available.” But when run under Windows 95, the program successfully allocates a 65th EMM handle, then adds it to the array, resulting in a buffer overflow. And just our luck, the variable stored immediately after the array holds the location of the EMM page frame.

As a result, when the program tries to copy memory into or out of the EMM page frame, it instead copies memory into and out of some random address. This quickly leads to scrambled memory and the program crashes spectacularly.

Of course, once you know what the problem is, the fix is simple: Limit the program to at most 64 EMM handles.

Raymond Chen

Follow