Is there a fixed virtual address that the system promises never to use for anything, so I can use it?

devblogs.microsoft.com/oldnewthing/20221222-00

December 22, 2022

Raymond Chen

A customer was interfacing with a component that required that shared memory blocks have the same virtual address in all processes.[1] The customer picked an address and tried to allocate memory at that address at the very start of the process, but they found that it would fail occasionally because address space layout randomization (ASLR) happened to choose that address for a DLL or heap or stack or *something*.

The customer wanted to know if there was some reserved region of address space that the operating system promises never to use, so that applications can use it as a hard-coded address.

And since everything comes in waves,[2] about four weeks later, a different customer asked basically the same question: They have been experiencing an increase in problems caused by the inability to allocate shared memory at the same virtual address in all client processes because they store pointers in the shared memory block. The code is decades old, ported from an earlier mainframe system, and none of the original developers are still around. This customer fully acknowledges that their design is subject to the vagaries of the process address space, and that the proper solution is to store offsets in their shared memory block instead of raw pointers. They're working on a complete rewrite of this module that will address the problem, but it's not quite finished yet and are hoping for some ideas that could buy some time and let them limp along with the original architecture until the rewrite is finished.

So how about it. Is there some fixed address we can safely reserve across all processes?

No, there is no such reserved region.

First of all, reserving such a region would impair ASLR, since it reduces the space of addresses that can be used for randomization.

Second, it creates the "What if two programs did this?" problem.

Suppose you had two programs, both of which claim the reserved region for themselves. And then they try to communicate with each other. The first one tries to allocate the matching address in the second process, but finds that it has already been taken. So it already has to deal with the possibility that the address it wants is unavailable in all processes.

The problem could even occur within one process. Your process uses two components, and both of those components try to allocate memory at the "guaranteed available" address. The first one will succeed and the second will fail.

So you still have the problem where any component has to deal with the possibility that the address it wants to use is not available, and develop a fallback. You haven't gained anything. You may as well just use the fallback all the time.

Bonus craziness: One idea is to create a DLL that consists of a large BSS section, at big as the size of the shared memory they intend to use. And then *strip the relocations from it*! Since the DLL is non-relocatable, it must load at its preferred base address, or it will never load at all. Have the primary executable contain a load-time link to this DLL, and that may help get the DLL loaded into memory ahead of other things that would normally consume the address space. If you mark the BSS section as shared, then you have a shared memory block at a fixed address. Of course, if that address is not available, then the process will fail to start. It's "get the address or die."

Note also that your shared section is now a security vulnerability, so this technique assumes that you can mitigate those security issues, say by running the program only under very controlled system configurations.

[1] My guess is that they store raw pointers directly in the shared memory block, and therefore also have the hidden requirement that all clients be the same bitness.

[2] Just like how two volcano disaster movies came out in 1997, or two animated insect movies came out in 1998, or two astronomical object threatens to destroy the planet movies came out in 1998.

Raymond Chen

Follow