Does mapping the same shared memory two times in a process lead to double the address space usage?

devblogs.microsoft.com/oldnewthing/20120127-00

January 27, 2012

Raymond Chen

A customer designed a system which uses shared memory. Specifically, for each database file, they create a corresponding shared memory block of, say, 200MB. Multiple clients which connect to the same database file use the same shared memory block. Naturally, if two processes each access the same database file, each process will map the shared memory block into their respective address space. The question arose regarding what happens if one process connects to the same database file twice. Will the two calls to `MapViewOfFile` share the same address space, or will each one allocate a separate chunk of address space? Win32 makes no guarantees what will happen. All that you can be sure of is that the memory will be mapped into your address space, and you will get a pointer to it, and when you're done, you call `UnmapViewOfFile`. Whether the two calls return the same pointer is unspecified. In fact, Windows 95 returned the same pointer, whereas Windows NT returns a different pointer. We saw this earlier when we intentionally mapped the same shared memory block multiple times, and observed somebody actually taking a dependency on this behavior in order to effect the strangest way of detecting Windows NT. Don't take a dependency on this behavior; who knows, maybe a future version of Windows NT will consolidate multiple mappings in order to conserve address space.

If you want force this consolidation behavior, you'll have to roll it yourself, say with a lookup table of active mappings and a reference count.

Raymond Chen

Follow