Is WriteProcessMemory atomic?

devblogs.microsoft.com/oldnewthing/20140515-00

May 15, 2014

Raymond Chen

A customer asked, "Does `WriteProcessMemory` write the memory atomically? I mean, if I use `WriteProcessMemory` to write 10 instructions for a total of 20 bytes, can `WriteProcessMemory` write those 20 bytes atomically?"

CPUs typically provide only modest atomic update capabilities. The x86 family of processors, for example, can update up to eight bytes atomically. Twenty bytes is beyond the capability of the processor.

I was kind of baffled at what sort of mental model of computing the customer had developed. It apparently permits `WriteProcessMemory` to accomplish something that the CPU is not physically capable of performing.

"Will my aluminum hammer withstand temperatures above 700C?"

Given that aluminum melts at 660C, it doesn't matter whether you make a hammer or a ladder or a scaffold. As long as you make it out of aluminum, it will melt at 660C because that's a fundamental property of aluminum.

The only thing I can think of is that the customer thought that maybe the kernel suspended all of the threads in the process, updated the memory, and then unfroze them all. It wouldn't be an atomic update in an absolute sense (somebody else doing a `ReadProcessMemory` might read an in-progress write), but it would be atomic from the viewpoint of the process being written to.

But no, the `WriteProcessMemory` function does no such thing. It merely writes the memory into the process address space.

Another way of thinking about it is using the thought experiment "Imagine if this were true." If it were true that `WriteProcessMemory` provided atomicity guarantees for 20 bytes, then all sorts of multi-threaded synchronization problems would magically disappear. If you wanted to update a block of memory in your process atomically, you would just call `WriteProcessMemory` on your own process handle!

I noted that the underlying scenario sounds really fishy. Using `WriteProcessMemory` to update code in a process sounds an awful lot like the customer is writing a virus. One of my colleagues who studies malware agreed, adding, “On the other hand, some anti-malware products also use that approach, as dubious as it is. For the record, I would like to add, ‘yuck’.” My colleague asked the customer for further details on what they are doing, and why they think that `WriteProcessMemory` is what they need, so that a proper solution to their underlying problem could be developed.

We never heard back from the customer.

Raymond Chen

Follow