How can I atomically leave a critical section and delete it?

devblogs.microsoft.com/oldnewthing/20170414-00

April 14, 2017

Raymond Chen

A customer had a thread which has entered a critical section. It has finished whatever it needs to do, and now it wants to leave the critical section and delete it. But wait, there's a race condition here: Between the call to `LeaveCriticalSection` and `DeleteCritical-Section`, another thread might sneak in and enter the critical section. This means that the original thread is going to delete a critical section while there are threads waiting for it, which the documentation explicitly calls out as leaving the waiting threads in an undefined state.

The question then is how to close this race window by performing the leave and delete atomically?

Actually, that's the wrong question, because having such an atomic operation doesn't fix anything. Suppose your thread calls that atomic leave-and-delete function. Now, the other interloper thread cannot enter the critical section after you leave and before you delete it. Yay, the race condition is gone!

But wait, you jumped out of the frying pan and landed in the fire: What's going to happen is that the interloper thread will instead try to enter the critical section *after it has been deleted*, which also results in undefined behavior.

All you did was trade one undefined behavior for another. You didn't actually fix anything.

If you have a system set up where there's the possibility of a thread entering a critical section that is about to be deleted, then it means that you also have the possibility of a thread entering a critical section *after* it has been deleted.

So fix your design so that second problem no longer exists. Once you fix that, you'll see that the original problem also vanishes.

Raymond Chen

Follow