What happens if I wake a condition variable when nobody is waiting for it? Is the wake saved for the next thread that waits?

devblogs.microsoft.com/oldnewthing/20170925-00

September 25, 2017

Raymond Chen

Suppose you call `WakeXxxConditionVariable` to wake a `CONDITION_VARIABLE`, but nobody is waiting on the condition variable. What happens? In particular, is the wake saved for the next thread that waits, so the next call to `SleepConditionVariableXxx` returns immediately instead of waiting?

The answer is, “It shouldn’t matter.”

The intended use pattern for a condition variable is to do the following:

1. Enter a lock.
2. Check a condition.
3. If the condition is false, then call `SleepConditionVariableXxx` and then go to step 2.
4. Perform an operation.
5. Release the lock.

And the code that establishes the condition (or at least changes the condition so it might be true for at least one waiter) calls `WakeXxxConditionVariable`.

If you follow this pattern, then it doesn’t matter whether a call to `WakeXxxConditionVariable` is remembered when there are no waiting threads. According to the intended use pattern, a thread is expected to check the condition first, and only if the condition is false should the thread call `SleepConditionVariableXxx`. Whether the wake is remember or not is irrelevant because the waiting thread never actually waits!

In other words, if you are counting on an unnecessary wake being saved and waking up a future sleep, then that means that you went to sleep before checking the condition. (Because if you had checked the condition, you would have avoided the sleep.) You’re holding it wrong.

Conversely, if you didn't expect the unnecessary wake to be remembered, but you got one anyway, well, that's also permitted because condition variables are explicitly documented as subject to spurious wakes. Again, if you follow the intended use pattern, spurious wakes aren't a problem (aside from performance) because the recommended pattern is to re-check the condition after the sleep wakes. If the wake were spurious, the check would fail, and you would go back to sleep.

In summary, if you wake a condition variable when nobody is waiting for it, it is unspecified whether the wake is saved for the next thread that waits, and that's okay, because if you follow the intended use pattern, it doesn't matter.

Raymond Chen

Follow