

During process termination, slim reader/writer locks are now also electrified

 devblogs.microsoft.com/oldnewthing/20111007-00

October 7, 2011



Raymond Chen

Some time ago I mentioned that during process termination, the gates are now electrified: If you attempt to enter a critical section that is owned by a thread that was terminated by an earlier phase of process termination, the entire process is forcibly terminated. Windows Vista introduced a new lightweight synchronization pseudo-object known as the slim reader/writer lock. And if you tried to enter a slim reader/writer lock during process termination and found yourself waiting for the current owner to release it, you ended up waiting forever since the current owner was terminated by an earlier phase of process termination. The sentence “As for the home-grown stuff, well, you’re on your own” applies here. Even though the slim reader/writer lock functions are exported from `kernel32.dll`, they don’t have any special kernel powers with respect to process termination. From the standpoint of process termination, they may as well be some home-grown synchronization primitive.

In Windows 7, the kernel folks decided to bring slim reader/writer locks into the fold of *objects which are electrified during process termination*. Starting in Windows 7, if you attempt to acquire a slim reader/writer lock during process termination, and the lock cannot be immediately acquired, then the process is forcibly terminated.

[Raymond Chen](#)

Follow

