

Why is the !locks command called !locks even though it debugs only critical sections?

devblogs.microsoft.com/oldnewthing/20170518-00

May 18, 2017



Raymond Chen

Commenter DWalker07 notes that the `!locks` command is called `!locks` even though it doesn't actually debug all types of locks, only critical sections. Why the bad name?

Because at the time it was written, critical sections were the only types of user-mode locks.

In the original incarnation of Win32, there were semaphores and mutexes and other kernel-mode synchronization objects, but the only user-mode synchronization objects were critical sections. Seeing as they were the only type of synchronization objects available in user mode, they were informally referred to as “locks” because when you have only one kind of thing, you don't really worry too much about being specific.

If there's only kind of wristwatch, then you don't really bother being specific about the fact that it's a spring-wound analog-readout time-only watch. You just call it a watch. Of course, later, when battery-powered digital-readout smart watches become commonplace, your old documents that talk about “watches” may seem confusing. “Why do these instructions for troubleshooting watches not work for my battery-powered digital-readout smart watch?”

Indeed, for a very long time, critical sections were the only user-mode synchronization objects that came with Win32. It wasn't until Windows Vista that a whole bunch of new user-mode synchronization objects became available, things like slim reader-writer locks, condition variables, and one-time initialization. Until then, the word “lock” was sufficient.

[Raymond Chen](#)

Follow

