

QueryPerformanceCounter counts elapsed time, not CPU cycles

devblogs.microsoft.com/oldnewthing/20080908-00

September 8, 2008



Raymond Chen

An anonymous coward asks whether the QueryPerformanceCounter function counts elapsed time or CPU cycles.

It counts elapsed time. It has to, since its value is governed by the `QueryPerformanceFrequency` function, which returns a number specifying the number of units per second, and the frequency is spec'd as not changing while the system is running.

For CPUs that can run at variable speed, this means that the HAL cannot use an instruction like `RDTSC`, since that does not correlate with elapsed time. Commenter "A" appears to have found a buggy HAL that failed to take this into account and returns values that do not correlate with elapsed time.

What would it take to create a counter that was tied to CPU cycles? Well, first, you'd have to come up with some definition of "CPU cycles" that is architecture-neutral. Maybe you'd say that it's a 64-bit value that increments at a rate proportional to the amount of work the CPU has done. And then you have to come up with some sort of definition of what should happen on multi-processor machines. What if you have two CPUs, one of which has gone into a `HLT` state (not running), while the other is busy doing work? Should the "cycle counter" run at half speed, since only half of the CPUs are running at full speed? What about hyperthreaded processors? It's all so confusing.

As a final remark, commenter Ulric wanted to know what I meant when I wrote, "Throw in a handful of workarounds for known buggy hardware." What I meant was that the people who write HALs are aware of various types of buggy hardware and added code to detect that buggy hardware and work around the problems.

Raymond Chen

Follow

