How accurate are the various Windows time-querying functions?

devblogs.microsoft.com/oldnewthing/20170921-00

September 21, 2017

Raymond Chen

Windows has a bunch of time-querying functions. One group of functions uses the system performance counter. These are as accurate as the hardware allows while still conforming to the basic requirements (such as running at a constant speed and being consistent among multiple processors).

- `QueryPerformanceCounter` gives you the current performance counter, and
- `QueryPerformanceFrequency` tells you the frequency of the performance counter.

Another group uses the system timer, which usually means 55ms or 10ms, although the `timeBeginPeriod` function can be used to run the timer at a higher rate.

- `GetTickCount` , `GetTickCount64`
- `GetMessageTime`
- `GetSystemTime` , `GetLocalTime` , `GetSystemTimeAsFileTime`
- `QueryInterruptTime` , `QueryUnbiasedInterruptTime`

And then there are the so-called precise[1] variants of the system timer functions. These take the system timer value and combine it with the system performance counter to get a high-accuracy timestamp. It's not only querying two timers, but it's also doing additional computations to combine the values, so naturally this is slower than querying just one of the two timers, but hey, a fine wine takes time.

- `GetSystemTimePreciseAsFileTime`
- `QueryInterruptTimePrecise` , `QueryUnbiasedInterruptTimePrecise`

These high-accuracy functions give you the best of both worlds: Time correlated with real-world clocks, but with the accuracy of the system performance counter. But as noted, it comes at a performance cost.

[1] Though to be pedantic, they should be called the accurate variants.

Raymond Chen

Follow