Precision is not the same as accuracy

devblogs.microsoft.com/oldnewthing/20050902-00

September 2, 2005

Raymond Chen

Accuracy is how close you are to the correct answer; precision is how much resolution you have for that answer.

Suppose you ask me, “What time is it?”

I look up at the sun, consider for a moment, and reply, “It is 10:35am and 22.131 seconds.”

I gave you a very precise answer, but not a very accurate one.

Meanwhile, you look at your watch, one of those fashionable watches with notches only at 3, 6, 9 and 12. You furrow your brow briefly and decide, “It is around 10:05.” Your answer is more accurate than mine, though less precise.

Now let’s apply that distinction to some of the time-related functions in Windows.

The `GetTickCount` function has a precision of one millisecond, but its accuracy is typically much worse, dependent on your timer tick rate, typically 10ms to 55ms. The `GetSystemTimeAsFileTime` function looks even more impressive with its 100-nanosecond precision, but its accuracy is not necessarily any better than that of `GetTickCount` .

If you’re looking for high accuracy, then you’d be better off playing around with the `QueryPerformanceCounter` function. You have to make some tradeoffs, however. For one, the precision of the result is variable; you need to call the `QueryPerformanceFrequency` function to see what the precision is. Another tradeoff is that the higher accuracy of `QueryPerformanceCounter` can be slower to obtain.

What `QueryPerformanceCounter` actually does is up to the HAL (with some help from ACPI). The performance folks tell me that, in the worst case, you might get it from the rollover interrupt on the programmable interrupt timer. This in turn may require a PCI transaction, which is not exactly the fastest thing in the world. It’s better than `GetTickCount` , but it’s not going to win any speed contests. In the best case, the HAL may conclude that the RDTSC counter runs at a constant frequency, so it uses that instead. Things are particularly exciting on multiprocessor machines, where you also have to make sure that the values returned from RDTSC on each processor are consistent with each other! And then, for good measure, throw in a handful of workarounds for known buggy hardware.

Raymond Chen

Follow