**LINUX TIMER HANDLING**

Author: Vũ Công Hoàn

CPU counters and wall clocks are different tools for different purposes.

**When to use a wall clock:**

When you want to measure time in a standard time unit (such as seconds). If you want to measure how long X task takes, use a wall clock.

Examples:

* clock()
* gettimeofday()
* clock\_gettime(2)
* etc...

**When to use RDTSC:**

If you're looking to measure the *relative* times of two different tasks to as high precision as possible, then RDTSC may be suitable.

RDTSC measures the number of pseudo-cycles that have elapsed since the CPU has started up. Often (but not always), this is equal to the CPU clock speed of your processor. But there's no easy to determine the exact number of "ticks per second" without actually measuring it against a wall clock.

However, RDTSC is about as low overhead as it can get for a time function. So it is well suited for micro-optimizations when you're comparing one implementation against another to determine which is *faster*. (as opposed to how much absolute time it takes)

Other things to note:

* In most cases, most benchmarking purposes can be done sufficiently well with wall clocks. So the use of RDTSC is pretty limited. Stick with standardized functions when possible.
* High precision wall clocks are typically implemented on top of RDTSC. So if you're trying to use RDTSC to get a high-precision measurement of wall time, you'll just be reinventing the wheel.