Re: Is there a problem in timeval_to_jiffies?

Source: http://linux.derkeiler.com/Mailing−Lists/Kernel/2004−09/5631.html

---

From: George Anzinger (*george_at_mvista.com*)
Date: 09/16/04

```
Date:   Thu, 16 Sep 2004 13:19:58 −0700
To: Henry Margies <henry.margies@gmx.de>, lkml <linux−kernel@vger.kernel.org>
```

Henry Margies wrote:
> Hi,
>
>
> On Thu, 16 Sep 2004 02:54:39 −0700
> George Anzinger <george@mvista.com> wrote:
>
>
>>Timers are constrained by the standard to NEVER finish early.
>
>
> I just thought about that again and I think you are wrong.
> Maybe your statement is true for one−shot timers, but not for
> interval timers.
>
> No interval timer can guarantee, that the time between to
> triggers is always greater or equal to the time you programmed
> it.

This depends on how you interpret things. Strictly speaking you are right in that a given timer signal can be delayed (latency things) while the next signal is not so that that interval would appear short. However, the standard seems to say that what you should measure is the expected arrival time (i.e. assume zero latency). In this case the standard calls for timers NEVER to be early.
>
> 1 occurrence of a 1000ms timer,
> 10 occurrences of a 100ms timer and
> 100 occurrences of a 10ms timer should take the same time.

You are assuming NICE things about timers that just are not true. The problem is resolution. The timer resolution is a function of what the hardware can actually do. The system code attempts to make the resolution as close to 1/HZ as possible, but this will not always be exact. In fact, the best that the x86 hardware can do with HZ=1000 is 999849 nanoseconds. Hence the result as per my message.
>
> For example:

> *> I want to have an interval timer for each second. Because of*
> *> some special reason the time between two triggers became 1.2*
> *> seconds.*
> *> The question is now, when do you want to have the next timer?*

You are talking about latency here. The kernel and the standard do not account for latency.

> *> Your approach would trigger the timer in at least one second. But*
> *> that is not the behavior of an interval timer. An interval timer*
> *> should trigger in 0.8 seconds because I wanted him to trigger*
> *> _every_ second.*

Yes, within the limits of the hardware imposed resolution.

> *> If you want to have at least one second between your timers, you*
> *> have to use one−shot timers and restart them after each*
> *> occurrence.*

Yes.

> *> And in fact, I think that no userspace program can ever take*
> *> advantage of your approach, because it can be interrupted*
> *> every time, so there is no guarantee at all, that there will be at*
> *> least some fixed time between the very important commands. (for*
> *> interval timers)*

Uh, my approach???

> *> So, what about adding this rounding value just to it_value to*
> *> guarantee that the first occurrence is in it least this time?*

The it_value and the it_interval are, indeed, computed differently. The it_value needs to have 1 additional resolution size period added to it to account for the initial time starting between ticks. The it_interval does not have this additional period added to it. Both values, however, are first rounded up to the next resolution size value.

−−
George Anzinger george@mvista.com
High−res−timers: http://sourceforge.net/projects/high−res−timers/
Preemption patch: http://www.kernel.org/pub/linux/kernel/people/rml
−