

# Re: [patch 02/23] GTOD: persistent clock support, core

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-09/msg09022.html>

---

- *From:* Andrew Morton <[akpm@xxxxxxxx](mailto:akpm@xxxxxxxx)>
  - *Date:* Sat, 30 Sep 2006 01:35:58 -0700
- 

On Fri, 29 Sep 2006 23:58:21 -0000  
Thomas Gleixner <[tglx@xxxxxxxxxxxxxx](mailto:tglx@xxxxxxxxxxxxxx)> wrote:

From: John Stultz <[johnstul@xxxxxxxxxxxx](mailto:johnstul@xxxxxxxxxxxx)>

persistent clock support: do proper timekeeping across suspend/resume.

How?

```
+/* Weak dummy function for arches that do not yet support it.  
+ * XXX - Do be sure to remove it once all arches implement it.  
+ */  
+unsigned long __attribute__((weak)) read_persistent_clock(void)  
+{  
+ return 0;  
+}
```

Seconds? microseconds? jiffies? walltime? uptime?

Needs some comments.

```
void __init timekeeping_init(void)  
{  
- unsigned long flags;  
+ unsigned long flags, sec = read_persistent_clock();
```

So it apparently returns seconds-since-epoch?

If so, why?

Re: [patch 02/23] GTOD: persistent clock support, core

```
write_seqlock_irqsave(&xtime_lock, flags);
```

```
@@ -758,11 +769,18 @@ void __init timekeeping_init(void)
clocksource_calculate_interval(clock, tick_nsec);
clock->cycle_last = clocksource_read(clock);
```

```
+ xtime.tv_sec = sec;
+ xtime.tv_nsec = (jiffies % HZ) * (NSEC_PER_SEC / HZ);
```

Why is it valid to take the second from the persistent clock and the fraction-of-a-second from jiffies? Some comments describing the implementation would improve its understandability and maintainability.

This statement can set `xtime.tv_nsec` to a value  $\geq$  `NSEC_PER_SEC`. Should it not be normalised?

```
+ set_normalized_timespec(&wall_to_monotonic,
+ -xtime.tv_sec, -xtime.tv_nsec);
+
write_sequnlock_irqrestore(&xtime_lock, flags);
}
```

```
static int timekeeping_suspended;
+static unsigned long timekeeping_suspend_time;
```

In what units?

```
+
/**
 * timekeeping_resume - Resumes the generic timekeeping subsystem.
 * @dev: unused
@@ -773,14 +791,23 @@ static int timekeeping_suspended;
 */
static int timekeeping_resume(struct sys_device *dev)
{
- unsigned long flags;
+ unsigned long flags, now = read_persistent_clock();
```

Would whoever keeps doing that please stop it? This:

```
unsigned long flags;
unsigned long now = read_persistent_clock();
```

is more readable and makes for more readable patches in the future.

Re: [patch 02/23] GTOD: persistent clock support, core

Re: [patch 02/23] GTOD: persistent clock support, core

```
write_seqlock_irqsave(&xtime_lock, flags);
- /* restart the last cycle value */
+
+ if (now && (now > timekeeping_suspend_time)) {
+ unsigned long sleep_length = now - timekeeping_suspend_time;
+ xtime.tv_sec += sleep_length;
+ jiffies_64 += sleep_length * HZ;
```

sleep\_length will overflow if we slept for more than 49 days, and HZ=1000.

```
+ }
+ /* re-base the last cycle value */
clock->cycle_last = clocksource_read(clock);
clock->error = 0;
timekeeping_suspended = 0;
write_sequnlock_irqrestore(&xtime_lock, flags);
+
+ hrtimer_notify_resume();
+
return 0;
}
```

```
@@ -790,6 +817,7 @@ static int timekeeping_suspend(struct sy
```

```
write_seqlock_irqsave(&xtime_lock, flags);
timekeeping_suspended = 1;
+ timekeeping_suspend_time = read_persistent_clock();
write_sequnlock_irqrestore(&xtime_lock, flags);
return 0;
}
```

--

-  
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in  
the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx  
More majordomo info at <http://vger.kernel.org/majordomo-info.html>  
Please read the FAQ at <http://www.tux.org/lkml/>