

# Developer's Diary: It's about time!

Wolfram Sang



28.10.2011, ELCE 2011

# Overview

1 Delays

2 Timeouts

# Basics

## Different principles of waiting

- 1 busy waiting
- 2 blocking

# Real life problem

Customer system used  $\approx 25\%$  CPU when pressing touch and otherwise idle.

# The delay

from drivers/input/touchscreen/wm9712.c<sup>1</sup>:

```
/*  
 * Delay after issuing a POLL command.  
 *  
 * The delay is 3 AC97 link frames + the touchpanel settling delay  
 */  
static inline void poll_delay(int d)  
{  
    udelay(3 * AC97_LINK_FRAME + delay_table[d]);  
}
```

---

<sup>1</sup>all code examples from Linux 3.1

# The setup

```
/*  
 * Set adc sample delay.  
 *  
 * For accurate touchpanel measurements, some settling time may be  
 * required between the switch matrix applying a voltage across the  
 * touchpanel plate and the ADC sampling the signal.  
 *  
 * This delay can be set by setting delay = n, where n is the array  
 * position of the delay in the array delay_table below.  
 * Long delays > 1ms are supported for completeness, but are not  
 * recommended.  
 */  
static int delay = 3;  
module_param(delay, int, 0);  
MODULE_PARM_DESC(delay, "Set adc sample delay.");
```

# The table!

```
/*  
 * ADC sample delay times in uS  
 */  
static const int delay_table[] = {  
    21,    /* 1 AC97 Link frames */  
    42,    /* 2 */  
    84,    /* 4 */  
    167,   /* 8 */  
    333,   /* 16 */  
    667,   /* 32 */  
    1000,  /* 48 */  
    1333,  /* 64 */  
    2000,  /* 96 */  
    2667,  /* 128 */  
    3333,  /* 160 */  
    4000,  /* 192 */  
    4667,  /* 224 */  
    5333,  /* 256 */  
    6000,  /* 288 */  
    0      /* No delay, switch matrix always on */  
};
```

# Any other delays like this?

Idea:

use ftrace to report delays



# A few challenges using ftrace directly

- trace `udelay` directly

On ARM: a define calling into an assembly function using two entry points

- function argument is convenient

not supported

- combine with `function_graph`

`function_graph` doesn't have `parent_ip` which was needed<sup>2</sup>

---

<sup>2</sup>sorry, forgot why, doh

# Keep it simple!

```

--- a/arch/arm/include/asm/delay.h
+++ b/arch/arm/include/asm/delay.h
@@ -34,11 +34,12 @@ extern void __const_udelay(unsigned long);

#define MAX_UDELAY_MS 2

-#define udelay(n) \
-    (__builtin_constant_p(n) ? \
-    ((n) > (MAX_UDELAY_MS * 1000) ? __bad_udelay() : \
-    __const_udelay((n) * ((2199023U*HZ)>>11))) : \
-    __udelay(n))
+static inline void udelay(unsigned long n)
+{
+    trace_printk("delays %lu\n", n);
+    (__builtin_constant_p(n) ? ((n) > (MAX_UDELAY_MS * 1000) ? __bad_udelay() :
+    __const_udelay((n) * ((2199023U*HZ)>>11))) : __udelay(n));
+}

#endif /* defined(_ARM_DELAY_H) */

```

# Typical output from a target

I'm brave: Demo time!

# Delays when waiting for power-up

from arch/arm/mach-mxs/module-tx28.c:

```
/* Power up fec phy */
pr_debug("%s: Switching FEC PHY power on\n", __func__);
ret = gpio_direction_output(TX28_FEC_PHY_POWER, 1);
if (ret) {
    pr_err("Failed to power on PHY: %d\n", ret);
    goto free_gpios;
}
mdelay(26); /* 25ms according to data sheet */
```

# Output from a target (with MMC)

Demo time again

# Finding mmc\_delay()

from drivers/mmc/core/core.h:

```
static inline void mmc_delay(unsigned int ms)
{
    if (ms < 1000 / HZ) {
        cond_resched();
        mdelay(ms);
    } else {
        msleep(ms);
    }
}
```

# Overview

1 Delays

2 Timeouts

# Timeout #0

from drivers/net/netx-eth.c:

```
static void
netx_eth_phy_write(struct net_device *ndev, int phy_id, int reg, int value)
{
    unsigned int val;

    val = MIIMU_SNRDY | MIIMU_PREAMBLE | MIIMU_PHYADDR(phy_id) |
          MIIMU_REGADDR(reg) | MIIMU_PHY_NRES | MIIMU_OPMODE_WRITE |
          MIIMU_DATA(value);

    writel(val, NETX_MIIMU);
    while (readl(NETX_MIIMU) & MIIMU_SNRDY);
}
```



# Timeout #1

from arch/arm/mach-mxs/clock-mx28.c:

```
for (i = 10000; i; i--)
    if (!(__raw_readl(CLKCTRL_BASE_ADDR +
        HW_CLKCTRL_HBUS) & BM_CLKCTRL_HBUS_ASM_BUSY))
        break;
if (!i) {
    pr_err("%s: divider writing timeout\n", __func__);
    return -ETIMEDOUT;
}
```

# Timeout #2

from drivers/mtd/onenand/onenand\_base.c:

```
timeout = jiffies + msecs_to_jiffies(20);
while (time_before(jiffies, timeout)) {
    interrupt = this->read_word(this->base + ONENAND_REG_INTERRUPT);

    if (interrupt & flags)
        break;

    if (state != FL_READING && state != FL_PREPARING_ERASE)
        cond_resched();
}
```

# Timeout #3

from drivers/media/dvb/dvb-core/dvb\_ca\_en50221.c:

```
timeout = jiffies + timeout_hz;
while (1) {
    /* read the status and check for error */
    int res = ca->pub->read_cam_control(ca->pub, slot, CTRLIF_STATUS);
    if (res < 0)
        return -EIO;
    /* if we got the flags, it was successful! */
    if (res & waitfor) {
        return 0;
    }
    /* check for timeout */
    if (time_after(jiffies, timeout)) {
        break;
    }
    /* wait for a bit */
    msleep(1);
}
/* if we get here, we've timed out */
return -ETIMEDOUT;
```

# Timeout #4

from drivers/misc/eeprom/at24.c:

```
timeout = jiffies + msecs_to_jiffies(write_timeout);
do {
    read_time = jiffies;

    switch (at24->use_smbus) {
        status = ...
    }

    if (status == count)
        return count;

    /* REVISIT: at HZ=100, this is sloooow */
    msleep(1);
} while (time_before(read_time, timeout));

return -ETIMEDOUT;
```

# Dummy slide

:D

# Dummy slide 2



# The End

Thank you for your attention!

## Questions? Comments?

- right now
- anytime at this conference
- [wsa@pengutronix.de](mailto:wsa@pengutronix.de)