
Subject: Re: [PATCH/RFC] kthread API conversion for dvb_frontend and av7110
Posted by [Cedric Le Goater](#) on Wed, 24 Jan 2007 15:47:49 GMT

[View Forum Message](#) <> [Reply to Message](#)

Cedric Le Goater wrote:

> Andrew de Quincey wrote:

>

>> Hi - the conversion looks good to me.. I can't really offer any more

>> constructive suggestions beyond what Cedric has already said.

>

> ok. so, should we just resend a refreshed version of the patch when 2.6.19

> comes out ?

>

>> Theres another thread in dvb_ca_en50221.c that could be converted as well

>> though, hint hint ;)

>

> ok ok :) i'll look at it ...

Here's a try. Compiles and boots but I have no hardware to test the patch :(

could we replace wait_event_interruptible_timeout() with wait_event_timeout() ? I don't see who would signal the thread.

thanks,

C.

Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>

```
drivers/media/dvb/dvb-core/dvb_ca_en50221.c | 59 ++++++++-----  
1 file changed, 18 insertions(+), 41 deletions(-)
```

Index: 2.6.20-rc4-mm1/drivers/media/dvb/dvb-core/dvb_ca_en50221.c

=====

--- 2.6.20-rc4-mm1.orig/drivers/media/dvb/dvb-core/dvb_ca_en50221.c

+++ 2.6.20-rc4-mm1/drivers/media/dvb/dvb-core/dvb_ca_en50221.c

@@ -37,6 +37,7 @@

#include <linux/delay.h>

#include <linux/spinlock.h>

#include <linux/sched.h>

+#include <linux/kthread.h>

#include "dvb_ca_en50221.h"

#include "dvb_ringbuffer.h"

@@ -140,14 +141,11 @@ struct dvb_ca_private {

wait_queue_head_t wait_queue;

```

/* PID of the monitoring thread */
- pid_t thread_pid;
+ struct task_struct* thread;

/* Wait queue used when shutting thread down */
wait_queue_head_t thread_queue;

- /* Flag indicating when thread should exit */
- unsigned int exit:1;
-
/* Flag indicating if the CA device is open */
unsigned int open:1;

@@ -916,8 +914,6 @@ static int dvb_ca_en50221_thread_should_
    ca->wakeup = 0;
    return 1;
}
- if (ca->exit)
- return 1;

return 0;
}
@@ -982,7 +978,6 @@ static void dvb_ca_en50221_thread_update
static int dvb_ca_en50221_thread(void *data)
{
    struct dvb_ca_private *ca = data;
- char name[15];
    int slot;
    int flags;
    int status;
@@ -991,28 +986,19 @@ static int dvb_ca_en50221_thread(void *d

    dprintk("%s\n", __FUNCTION__);

- /* setup kernel thread */
- snprintf(name, sizeof(name), "kdvb-ca-%i:%i", ca->dvbdev->adapter->num, ca->dvbdev->id);
-
- lock_kernel();
- daemonize(name);
- sigfillset(&current->blocked);
- unlock_kernel();
-
/* choose the correct initial delay */
dvb_ca_en50221_thread_update_delay(ca);

/* main loop */
- while (!ca->exit) {

```

```

+ while (1) {
  /* sleep for a bit */
  if (!ca->wakeup) {
-   flags = wait_event_interruptible_timeout(ca->thread_queue,
-     dvb_ca_en50221_thread_should_wakeup(ca),
-     ca->delay);
-   if ((flags == -ERESTARTSYS) || ca->exit) {
-     /* got signal or quitting */
+   flags = wait_event_interruptible_timeout(
+     ca->thread_queue,
+     dvb_ca_en50221_thread_should_wakeup(ca) || kthread_should_stop(),
+     ca->delay);
+   if ((flags == -ERESTARTSYS) || kthread_should_stop())
      break;
-   }
  }
  ca->wakeup = 0;

```

```

@@ -1182,9 +1168,8 @@ static int dvb_ca_en50221_thread(void *d
}

```

```

/* completed */
- ca->thread_pid = 0;
+ ca->thread = NULL;
  mb();
- wake_up_interruptible(&ca->thread_queue);
  return 0;
}

```

```

@@ -1663,6 +1648,7 @@ int dvb_ca_en50221_init(struct dvb_adapt
int ret;
struct dvb_ca_private *ca = NULL;
int i;
+ struct task_struct *thread;

dprintk("%s\n", __FUNCTION__);

```

```

@@ -1682,9 +1668,8 @@ int dvb_ca_en50221_init(struct dvb_adapt
goto error;
}
init_waitqueue_head(&ca->wait_queue);
- ca->thread_pid = 0;
+ ca->thread = NULL;
init_waitqueue_head(&ca->thread_queue);
- ca->exit = 0;
ca->open = 0;
ca->wakeup = 0;
ca->next_read_slot = 0;

```

```

@@ -1711,13 +1696,14 @@ int dvb_ca_en50221_init(struct dvb_adapt

/* create a kthread for monitoring this CA device */

- ret = kernel_thread(dvb_ca_en50221_thread, ca, 0);
-
- if (ret < 0) {
+ thread = kthread_run(dvb_ca_en50221_thread, ca, "kdvb-ca-%i:%i",
+   ca->dvbdev->adapter->num, ca->dvbdev->id);
+ if (IS_ERR(thread)) {
+   ret = PTR_ERR(thread);
+   printk("dvb_ca_init: failed to start kernel_thread (%d)\n", ret);
+   goto error;
+ }
- ca->thread_pid = ret;
+ ca->thread = thread;
+   return 0;

error:
@@ -1748,17 +1734,8 @@ void dvb_ca_en50221_release(struct dvb_c
dprintfk("%s\n", __FUNCTION__);

/* shutdown the thread if there was one */
- if (ca->thread_pid) {
-   if (kill_proc(ca->thread_pid, 0, 1) == -ESRCH) {
-     printk("dvb_ca_release adapter %d: thread PID %d already died\n",
-       ca->dvbdev->adapter->num, ca->thread_pid);
-   } else {
-     ca->exit = 1;
-     mb();
-     dvb_ca_en50221_thread_wakeup(ca);
-     wait_event_interruptible(ca->thread_queue, ca->thread_pid == 0);
-   }
- }
+ if (ca->thread)
+   kthread_stop(ca->thread);

for (i = 0; i < ca->slot_count; i++) {
  dvb_ca_en50221_slot_shutdown(ca, i);

```

Containers mailing list
Containers@lists.osdl.org
<https://lists.osdl.org/mailman/listinfo/containers>
