
Subject: [PATCH/RFC] kthread API conversion for dvb_frontend and av7110
Posted by [Herbert Poetzl](#) on Thu, 14 Sep 2006 20:01:03 GMT

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

Okay, as I promised, I had a first shot at the dvb kernel_thread to kthread API port, and here is the result, which is running fine here since yesterday, including module load/unload and software suspend (which doesn't work as expected with or without this patch :), I didn't convert the dvb_ca_en50221 as I do not have such an interface, but if the conversion process is fine with the v4l-dvb maintainers, it should not be a problem to send a patch for that too ...

best,
Herbert

Signed-off-by: Herbert Poetzl <herbert@13thfloor.at>

```
diff -NurP linux-2.6.18-rc6/drivers/media/dvb/dvb-core/dvb_frontend.c
linux-2.6.18-rc6-kthread.v02.3/drivers/media/dvb/dvb-core/dvb_frontend.c
--- linux-2.6.18-rc6/drivers/media/dvb/dvb-core/dvb_frontend.c 2006-09-12 18:16:12 +0200
+++ linux-2.6.18-rc6-kthread.v02.3/drivers/media/dvb/dvb-core/dvb_frontend.c 2006-09-14
21:23:37 +0200
@@ -36,6 +36,7 @@
#include <linux/list.h>
#include <linux/suspend.h>
#include <linux/jiffies.h>
+#include <linux/kthread.h>
#include <asm/processor.h>

#include "dvb_frontend.h"
@@ -100,7 +101,7 @@ struct dvb_frontend_private {
    struct semaphore sem;
    struct list_head list_head;
    wait_queue_head_t wait_queue;
- pid_t thread_pid;
+ struct task_struct *thread;
    unsigned long release_jiffies;
    unsigned int exit;
    unsigned int wakeup;
@@ -508,19 +509,11 @@ static int dvb_frontend_thread(void *dat
    struct dvb_frontend *fe = data;
    struct dvb_frontend_private *fepriv = fe->frontend_priv;
    unsigned long timeout;
- char name [15];
    fe_status_t s;
```

```

struct dvb_frontend_parameters *params;

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

- snprintf (name, sizeof(name), "kdvb-fe-%i", fe->dvb->num);
-
- lock_kernel();
- daemonize(name);
- sigfillset(&current->blocked);
- unlock_kernel();
-
fepriv->check_wrapped = 0;
fepriv->quality = 0;
fepriv->delay = 3*HZ;
@@ -534,14 +527,16 @@ static int dvb_frontend_thread(void *dat
    up(&fepriv->sem);    /* is locked when we enter the thread... */

    timeout = wait_event_interruptible_timeout(fepriv->wait_queue,
-        dvb_frontend_should_wakeup(fe),
-        fepriv->delay);
- if (0 != dvb_frontend_is_exiting(fe)) {
+ dvb_frontend_should_wakeup(fe) || kthread_should_stop(),
+ fepriv->delay);
+
+ if (kthread_should_stop() || dvb_frontend_is_exiting(fe)) {
    /* got signal or quitting */
    break;
}

- try_to_freeze();
+ if (try_to_freeze())
+ continue;

    if (down_interruptible(&fepriv->sem))
        break;
@@ -591,7 +586,7 @@ static int dvb_frontend_thread(void *dat
    fe->ops.sleep(fe);
}

- fepriv->thread_pid = 0;
+ fepriv->thread = NULL;
mb();

dvb_frontend_wakeup(fe);
@@ -600,7 +595,6 @@ static int dvb_frontend_thread(void *dat

static void dvb_frontend_stop(struct dvb_frontend *fe)
{

```

```

- unsigned long ret;
  struct dvb_frontend_private *fepriv = fe->frontend_priv;

  dprintk ("%s\n", __FUNCTION__);
@@ -608,33 +602,17 @@ static void dvb_frontend_stop(struct dvb
  fepriv->exit = 1;
  mb();

- if (!fepriv->thread_pid)
- return;
-
- /* check if the thread is really alive */
- if (kill_proc(fepriv->thread_pid, 0, 1) == -ESRCH) {
- printk("dvb_frontend_stop: thread PID %d already died\n",
- fepriv->thread_pid);
- /* make sure the mutex was not held by the thread */
- init_MUTEX (&fepriv->sem);
+ if (!fepriv->thread)
  return;
- }
-
- /* wake up the frontend thread, so it notices that fe->exit == 1 */
- dvb_frontend_wakeup(fe);

- /* wait until the frontend thread has exited */
- ret = wait_event_interruptible(fepriv->wait_queue, 0 == fepriv->thread_pid);
- if (-ERESTARTSYS != ret) {
- fepriv->state = FESTATE_IDLE;
- return;
- }
+ kthread_stop(fepriv->thread);
+ init_MUTEX (&fepriv->sem);
  fepriv->state = FESTATE_IDLE;

  /* paranoia check in case a signal arrived */
- if (fepriv->thread_pid)
- printk("dvb_frontend_stop: warning: thread PID %d won't exit\n",
- fepriv->thread_pid);
+ if (fepriv->thread)
+ printk("dvb_frontend_stop: warning: thread %p won't exit\n",
+ fepriv->thread);
  }

s32 timeval_usec_diff(struct timeval lasttime, struct timeval curtime)
@@ -684,10 +662,11 @@ static int dvb_frontend_start(struct dvb
{
  int ret;
  struct dvb_frontend_private *fepriv = fe->frontend_priv;

```

```

+ struct task_struct *fe_thread;

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

- if (fepriv->thread_pid) {
+ if (fepriv->thread) {
    if (!fepriv->exit)
        return 0;
    else
@@ -701,18 +680,18 @@ static int dvb_frontend_start(struct dvb

    fepriv->state = FESTATE_IDLE;
    fepriv->exit = 0;
- fepriv->thread_pid = 0;
+ fepriv->thread = NULL;
    mb();

- ret = kernel_thread (dvb_frontend_thread, fe, 0);
-
- if (ret < 0) {
-     printk("dvb_frontend_start: failed to start kernel_thread (%d)\n", ret);
+ fe_thread = kthread_run(dvb_frontend_thread, fe,
+     "kdvb-fe-%i", fe->dvb->num);
+ if (IS_ERR(fe_thread)) {
+     ret = PTR_ERR(fe_thread);
+     printk("dvb_frontend_start: failed to start kthread (%d)\n", ret);
        up(&fepriv->sem);
        return ret;
    }
- fepriv->thread_pid = ret;
-
+ fepriv->thread = fe_thread;
    return 0;
}

diff -NurpP linux-2.6.18-rc6/drivers/media/dvb/ttpci/av7110.c
linux-2.6.18-rc6-kthread.v02.3/drivers/media/dvb/ttpci/av7110.c
--- linux-2.6.18-rc6/drivers/media/dvb/ttpci/av7110.c 2006-09-12 18:16:13 +0200
+++ linux-2.6.18-rc6-kthread.v02.3/drivers/media/dvb/ttpci/av7110.c 2006-09-14 21:21:03 +0200
@@ -51,6 +51,7 @@
#include <linux/firmware.h>
#include <linux/crc32.h>
#include <linux/i2c.h>
+ #include <linux/kthread.h>

#include <asm/system.h>

@@ -223,11 +224,10 @@ static void recover_arm(struct av7110 *a

```

```

static void av7110_arm_sync(struct av7110 *av7110)
{
- av7110->arm_rmmod = 1;
- wake_up_interruptible(&av7110->arm_wait);
+ if (av7110->arm_thread)
+ kthread_stop(av7110->arm_thread);

- while (av7110->arm_thread)
- msleep(1);
+ av7110->arm_thread = NULL;
}

static int arm_thread(void *data)
@@ -238,17 +238,11 @@ static int arm_thread(void *data)

dprintk(4, "%p\n",av7110);

- lock_kernel();
- daemonize("arm_mon");
- sigfillset(&current->blocked);
- unlock_kernel();
-
- av7110->arm_thread = current;
-
for (;;) {
    timeout = wait_event_interruptible_timeout(av7110->arm_wait,
-        av7110->arm_rmmod, 5 * HZ);
- if (-ERESTARTSYS == timeout || av7110->arm_rmmod) {
+ kthread_should_stop(), 5 * HZ);
+
+ if (-ERESTARTSYS == timeout || kthread_should_stop()) {
    /* got signal or told to quit*/
    break;
}
}
@@ -276,7 +270,6 @@ static int arm_thread(void *data)
    av7110->arm_errors = 0;
}

- av7110->arm_thread = NULL;
return 0;
}

@@ -2334,6 +2327,7 @@ static int __devinit av7110_attach(struc
const int length = TS_WIDTH * TS_HEIGHT;
struct pci_dev *pdev = dev->pci;
struct av7110 *av7110;
+ struct task_struct *thread;

```

```

int ret, count = 0;

dprintk(4, "dev: %p\n", dev);
@@ -2618,9 +2612,12 @@ static int __devinit av7110_attach(struc
    printk ("dvb-ttpci: Warning, firmware version 0x%04x is too old. "
        "System might be unstable!\n", FW_VERSION(av7110->arm_app));

- ret = kernel_thread(thread, (void *) av7110, 0);
- if (ret < 0)
+ thread = kthread_run(thread, (void *) av7110, "arm_mon");
+ if (IS_ERR(thread)) {
+ ret = PTR_ERR(thread);
+ goto err_stop_arm_9;
+ }
+ av7110->arm_thread = thread;

/* set initial volume in mixer struct */
av7110->mixer.volume_left = volume;
diff -Nurp linux-2.6.18-rc6/drivers/media/dvb/ttpci/av7110.h
linux-2.6.18-rc6-kthread.v02.3/drivers/media/dvb/ttpci/av7110.h
--- linux-2.6.18-rc6/drivers/media/dvb/ttpci/av7110.h 2006-09-12 18:16:13 +0200
+++ linux-2.6.18-rc6-kthread.v02.3/drivers/media/dvb/ttpci/av7110.h 2006-09-14 21:21:03 +0200
@@ -205,7 +205,6 @@ struct av7110 {
    struct task_struct *arm_thread;
    wait_queue_head_t arm_wait;
    u16 arm_loops;
- int arm_rmmod;

void *debi_virt;
dma_addr_t debi_bus;

```

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