

---

Subject: Re: [PATCH] kthread: saa7134-tvaudio.c  
Posted by [Andrew Morton](#) on Tue, 29 Aug 2006 21:39:02 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Tue, 29 Aug 2006 14:15:55 -0700  
Sukadev Bhattiprolu <sukadev@us.ibm.com> wrote:

>  
> Replace kernel\_thread() with kthread\_run() since kernel\_thread()  
> is deprecated in drivers/modules.  
>  
> Note that this driver, like a few others, allows SIGTERM. Not  
> sure if that is affected by conversion to kthread. Appreciate  
> any comments on that.  
>

hm, I think this driver needs more help.

- It shouldn't be using signals at all, really. Signals are for userspace IPC. The kernel internally has better/richer/faster/tighter ways of inter-thread communication.
- saa7134\_tvaudio\_fini()-versus-tvaudio\_sleep() looks racy:

```
if (dev->thread.scan1 == dev->thread.scan2 && !dev->thread.shutdown) {  
    if (timeout < 0) {  
        set_current_state(TASK_INTERRUPTIBLE);  
        schedule();  
    }
```

If the wakeup happens after the test of dev->thread.shutdown, that sleep will be permanent.

So in general, yes, the driver should be converted to the kthread API - this is a requirement for virtualisation, but I forget why, and that's the "standard" way of doing it.

- The signal stuff should go away if at all possible.
- the thread.shutdown field should go away and be replaced by kthread\_should\_stop().
- the tvaudio\_sleep() race might need some attention (simply moving the set\_current\_state() to before the add\_wait\_queue() will suffice).
- the complete\_and\_exit() stuff might (should) no longer be needed - kthread\_stop() does that.

Sorry ;)

```
> 2 files changed, 17 insertions(+), 20 deletions(-)
>
> Index: lx26-18-rc5/drivers/media/video/saa7134/saa7134.h
> =====
> --- lx26-18-rc5.orig/drivers/media/video/saa7134/saa7134.h 2006-08-29 14:02:44.000000000
-0700
> +++ lx26-18-rc5/drivers/media/video/saa7134/saa7134.h 2006-08-29 14:04:21.000000000
-0700
> @@ -311,10 +311,8 @@ struct saa7134_pgtable {
>
> /* tvaudio thread status */
> struct saa7134_thread {
> - pid_t                pid;
> - struct completion    exit;
> + struct task_struct *  task;
>   wait_queue_head_t    wq;
> - unsigned int         shutdown;
>   unsigned int         scan1;
>   unsigned int         scan2;
>   unsigned int         mode;
> Index: lx26-18-rc5/drivers/media/video/saa7134/saa7134-tvaudio.c
> =====
> --- lx26-18-rc5.orig/drivers/media/video/saa7134/saa7134-tvaudio.c 2006-08-29
14:02:44.000000000 -0700
> +++ lx26-18-rc5/drivers/media/video/saa7134/saa7134-tvaudio.c 2006-08-29
14:06:24.000000000 -0700
> @@ -28,6 +28,7 @@
> #include <linux/slab.h>
> #include <linux/delay.h>
> #include <linux/smp_lock.h>
> +#include <linux/kthread.h>
> #include <asm/div64.h>
>
> #include "saa7134-reg.h"
> @@ -357,7 +358,7 @@ static int tvaudio_sleep(struct saa7134_
> DECLARE_WAITQUEUE(wait, current);
>
> add_wait_queue(&dev->thread.wq, &wait);
> - if (dev->thread.scan1 == dev->thread.scan2 && !dev->thread.shutdown) {
> + if (dev->thread.scan1 == dev->thread.scan2 && !kthread_should_stop()) {
>   if (timeout < 0) {
>     set_current_state(TASK_INTERRUPTIBLE);
>     schedule();
> @@ -525,7 +526,7 @@ static int tvaudio_thread(void *data)
> allow_signal(SIGTERM);
> for (;;) {
```

```

> tvaudio_sleep(dev,-1);
> - if (dev->thread.shutdown || signal_pending(current))
> + if (kthread_should_stop() || signal_pending(current))
>     goto done;
>
> restart:
> @@ -633,7 +634,7 @@ static int tvaudio_thread(void *data)
>     for (;;) {
>         if (tvaudio_sleep(dev,5000))
>             goto restart;
> - if (dev->thread.shutdown || signal_pending(current))
> + if (kthread_should_stop() || signal_pending(current))
>         break;
>         if (UNSET == dev->thread.mode) {
>             rx = tvaudio_getstereo(dev,&tvaudio[i]);
> @@ -649,7 +650,6 @@ static int tvaudio_thread(void *data)
>     }
>
> done:
> - complete_and_exit(&dev->thread.exit, 0);
>     return 0;
> }
>
> @@ -798,7 +798,6 @@ static int tvaudio_thread_ddep(void *dat
>     struct saa7134_dev *dev = data;
>     u32 value, norms, clock;
>
> - daemonize("%s", dev->name);
>     allow_signal(SIGTERM);
>
>     clock = saa7134_boards[dev->board].audio_clock;
> @@ -812,7 +811,7 @@ static int tvaudio_thread_ddep(void *dat
>
>     for (;;) {
>         tvaudio_sleep(dev,-1);
> - if (dev->thread.shutdown || signal_pending(current))
> + if (kthread_should_stop() || signal_pending(current))
>         goto done;
>
>     restart:
> @@ -894,7 +893,6 @@ static int tvaudio_thread_ddep(void *dat
>     }
>
> done:
> - complete_and_exit(&dev->thread.exit, 0);
>     return 0;
> }
>

```

```

> @@ -1004,15 +1002,16 @@ int saa7134_tvaudio_init2(struct saa7134
> break;
> }
>
> - dev->thread.pid = -1;
> + dev->thread.task = NULL;
> if (my_thread) {
> /* start tvaudio thread */
> init_waitqueue_head(&dev->thread.wq);
> - init_completion(&dev->thread.exit);
> - dev->thread.pid = kernel_thread(my_thread,dev,0);
> - if (dev->thread.pid < 0)
> + dev->thread.task = kthread_run(my_thread,dev,dev->name);
> + if (IS_ERR(dev->thread.task)) {
> printk(KERN_WARNING "%s: kernel_thread() failed\n",
> - dev->name);
> + dev->name);
> + dev->thread.task = NULL;
> + }
> saa7134_tvaudio_do_scan(dev);
> }
>
> @@ -1023,10 +1022,10 @@ int saa7134_tvaudio_init2(struct saa7134
> int saa7134_tvaudio_fini(struct saa7134_dev *dev)
> {
> /* shutdown tvaudio thread */
> - if (dev->thread.pid >= 0) {
> - dev->thread.shutdown = 1;
> - wake_up_interruptible(&dev->thread.wq);
> - wait_for_completion(&dev->thread.exit);
> + if (dev->thread.task) {
> + /* kthread_stop() wakes up the thread */
> + kthread_stop(dev->thread.task);
> + dev->thread.task = NULL;
> }
> saa_andorb(SAA7134_ANALOG_IO_SELECT, 0x07, 0x00); /* LINE1 */
> return 0;
> @@ -1034,7 +1033,7 @@ int saa7134_tvaudio_fini(struct saa7134_
>
> int saa7134_tvaudio_do_scan(struct saa7134_dev *dev)
> {
> - if (dev->thread.pid >= 0) {
> + if (dev->thread.task) {
> dev->thread.mode = UNSET;
> dev->thread.scan2++;
> wake_up_interruptible(&dev->thread.wq);

```

---

Containers mailing list

