
Subject: Re: [PATCH] saa7134-tvaudio: Convert to kthread API.
Posted by [Cedric Le Goater](#) on Fri, 20 Apr 2007 12:48:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Eric W. Biederman wrote:

> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
>
> It is my goal to replace all kernel code that handles signals
> from user space, calls kernel_thread or calls daemonize. All
> of which the kthread_api makes unnecessary. Handling signals
> from user space is a maintenance problem because using a
> kernel thread is an implementation detail and if user space
> cares it does not allow us to change the implementation. Calling
> daemonize is a problem because it has to undo a continually changing
> set of state generated by user space, requiring the implementation
> to change continually. kernel_thread is a problem because it
> returns a pid_t value. Numeric pids are inherently racy and
> in the presence of a pid namespace they are no longer global
> making them useless for general use in the kernel.
>
> So this patch renames the pid member of struct saa7134_thread
> started and changes its type from pid_t to int. All it
> has ever been used for is to detect if the kernel thread
> is has been started so this works.
>
> allow_signal(SIGTERM) and the calls to signal_pending have
> been removed they are needed for the driver to operation.
>
> The startup of tvaudio_thread and tvaudio_thread_dep have
> been modified to use kthread_run instead of a combination
> of kernel_thread and daemonize.
>
> The result is code that is slightly simpler and more
> maintainable.

Here's a refreshed attempt using kthread_should_stop().
Unfortunately, not tested bc we don't have the hardware.

cheers,

C.

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>

Replace kernel_thread() with kthread_run() since kernel_thread()
is deprecated in drivers/modules. Also remove signalling code
as it is not needed in the driver.

Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>
Cc: Mauro Carvalho Chehab <mchehab@infradead.org>
Cc: Containers@lists.osdl.org
Cc: video4linux-list@redhat.com
Cc: v4l-dvb-maintainer@linuxtv.org

drivers/media/video/saa7134/saa7134-tvaudio.c | 45 ++++++++-----
drivers/media/video/saa7134/saa7134.h | 4 --
2 files changed, 24 insertions(+), 25 deletions(-)

Index: 2.6.21-rc6-mm1/drivers/media/video/saa7134/saa7134.h

--- 2.6.21-rc6-mm1.orig/drivers/media/video/saa7134/saa7134.h
+++ 2.6.21-rc6-mm1/drivers/media/video/saa7134/saa7134.h
@@ -324,10 +324,8 @@ 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: 2.6.21-rc6-mm1/drivers/media/video/saa7134/saa7134-tvaudio.c

--- 2.6.21-rc6-mm1.orig/drivers/media/video/saa7134/saa7134-tvaudio.c
+++ 2.6.21-rc6-mm1/drivers/media/video/saa7134/saa7134-tvaudio.c
@@ -27,6 +27,7 @@
#include <linux/kernel.h>
#include <linux/slab.h>
#include <linux/delay.h>
+#include <linux/kthread.h>
#include <asm/div64.h>

#include "saa7134-reg.h"
@@ -344,16 +345,22 @@ 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) {
+
+ set_current_state(TASK_INTERRUPTIBLE);
+

```

+ if (dev->thread.scan1 == dev->thread.scan2 && !kthread_should_stop()) {
    if (timeout < 0) {
- set_current_state(TASK_INTERRUPTIBLE);
    schedule();
    } else {
        schedule_timeout_interruptible
            (msecs_to_jiffies(timeout));
    }
}
+
+ set_current_state(TASK_RUNNING);
+
+ remove_wait_queue(&dev->thread.wq, &wait);
+
+ return dev->thread.scan1 != dev->thread.scan2;
}

```

```

@@ -505,11 +512,9 @@ static int tvaudio_thread(void *data)
    unsigned int i, audio, nscan;
    int max1,max2,carrier,rx,mode,lastmode,default_carrier;

```

```

- daemonize("%s", dev->name);
- allow_signal(SIGTERM);
    for (;;) {
        tvaudio_sleep(dev,-1);
- if (dev->thread.shutdown || signal_pending(current))
+ if (kthread_should_stop())
        goto done;

```

restart:

```

@@ -618,7 +623,7 @@ 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())
        break;
        if (UNSET == dev->thread.mode) {
            rx = tvaudio_getstereo(dev,&tvaudio[i]);
@@ -634,7 +639,6 @@ static int tvaudio_thread(void *data)
    }

```

done:

```

- complete_and_exit(&dev->thread.exit, 0);
    return 0;
}

```

```

@@ -782,9 +786,6 @@ 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;
if (UNSET != audio_clock_override)
    clock = audio_clock_override;
@@ -796,7 +797,7 @@ static int tvaudio_thread_ddep(void *dat

for (;;) {
    tvaudio_sleep(dev,-1);
- if (dev->thread.shutdown || signal_pending(current))
+ if (kthread_should_stop())
    goto done;

restart:
@@ -876,7 +877,6 @@ static int tvaudio_thread_ddep(void *dat
}

done:
- complete_and_exit(&dev->thread.exit, 0);
return 0;
}

@@ -986,15 +986,16 @@ 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)
-     printk(KERN_WARNING "%s: kernel_thread() failed\n",
+ dev->thread.task = kthread_run(my_thread, dev, dev->name);
+ if (IS_ERR(dev->thread.task)) {
+     printk(KERN_WARNING "%s: failed to create kthread\n",
        dev->name);
+     dev->thread.task = NULL;
+ }
    saa7134_tvaudio_do_scan(dev);
}

```

```

@@ -1005,10 +1006,10 @@ 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;
@@ -1020,7 +1021,7 @@ int saa7134_tvaudio_do_scan(struct saa71
    dprintk("sound IF not in use, skipping scan\n");
    dev->automute = 0;
    saa7134_tvaudio_setmute(dev);
- } else if (dev->thread.pid >= 0) {
+ } else if (dev->thread.task) {
    dev->thread.mode = UNSET;
    dev->thread.scan2++;
    wake_up_interruptible(&dev->thread.wq);

```

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>
