
Subject: [PATCH] dvb_en_50221: Convert to kthread API
Posted by [ebiederm](#) on Thu, 19 Apr 2007 07:59:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

From: Eric W. Biederman <ebiederm@xmission.com>

This patch is a minimal transformation to use the kthread API doing it's best to preserve the existing logic.

Instead of starting kdvb-ca by calling kernel_thread, daemonize and sigfillset we kthread_run is used.

Instead of tracking the pid of the running thread we instead simply keep a flag to indicate that the current thread is running, as that is all the pid is really used for.

And finally the kill_proc sending signal 0 to the kernel thread to ensure it is alive before we wait for it to shutdown is removed. The kthread API does not provide the pid so we don't have that information readily available and the test is just silly. If there is no shutdown race the test is a useless confirmation of that the thread is running. If there is a race the test doesn't fix it and we should fix the race properly.

Cc: Andrew de Quincey <adq_dvb@lidskialf.net>
Cc: Mauro Carvalho Chehab <mchehab@infradead.org>
Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

drivers/media/dvb/dvb-core/dvb_ca_en50221.c | 46 ++++++++-----
1 files changed, 18 insertions(+), 28 deletions(-)

diff --git a/drivers/media/dvb/dvb-core/dvb_ca_en50221.c
b/drivers/media/dvb/dvb-core/dvb_ca_en50221.c
index 2a03bf5..b28bc15 100644

--- a/drivers/media/dvb/dvb-core/dvb_ca_en50221.c
+++ b/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"
@@ -139,8 +140,8 @@ struct dvb_ca_private {
/* wait queues for read() and write() operations */
wait_queue_head_t wait_queue;

```

- /* PID of the monitoring thread */
- pid_t thread_pid;
+ /* Flag indicating the monitoring thread is running */
+ int thread_running;

/* Wait queue used when shutting thread down */
wait_queue_head_t thread_queue;
@@ -982,7 +983,6 @@ static void dvb_ca_en50221_thread_update_delay(struct dvb_ca_private
*ca)
static int dvb_ca_en50221_thread(void *data)
{
    struct dvb_ca_private *ca = data;
- char name[15];
    int slot;
    int flags;
    int status;
@@ -991,14 +991,6 @@ static int dvb_ca_en50221_thread(void *data)

    dprintf("%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);

@@ -1182,7 +1174,7 @@ static int dvb_ca_en50221_thread(void *data)
}

/* completed */
- ca->thread_pid = 0;
+ ca->thread_running = 0;
    mb();
    wake_up_interruptible(&ca->thread_queue);
    return 0;
@@ -1660,6 +1652,7 @@ static struct dvb_device dvbdev_ca = {
int dvb_ca_en50221_init(struct dvb_adapter *dvb_adapter,
    struct dvb_ca_en50221 *pubca, int flags, int slot_count)
{
+ struct task_struct *task;
    int ret;
    struct dvb_ca_private *ca = NULL;
    int i;

```

```

@@ -1682,7 +1675,7 @@ int dvb_ca_en50221_init(struct dvb_adapter *dvb_adapter,
    goto error;
}
    init_waitqueue_head(&ca->wait_queue);
- ca->thread_pid = 0;
+ ca->thread_running = 0;
    init_waitqueue_head(&ca->thread_queue);
    ca->exit = 0;
    ca->open = 0;
@@ -1711,13 +1704,15 @@ int dvb_ca_en50221_init(struct dvb_adapter *dvb_adapter,

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

- ret = kernel_thread(dvb_ca_en50221_thread, ca, 0);
-
- if (ret < 0) {
-     printk("dvb_ca_init: failed to start kernel_thread (%d)\n", ret);
+ task = kthread_run(dvb_ca_en50221_thread, ca,
+     "kdvb-ca-%i:%i",
+     ca->dvbdev->adapter->num, ca->dvbdev->id);
+ if (IS_ERR(task)) {
+     ret = PTR_ERR(task);
+     printk("dvb_ca_init: failed to start kthread (%d)\n", ret);
        goto error;
    }
- ca->thread_pid = ret;
+ ca->thread_running = 1;
    return 0;

error:
@@ -1748,16 +1743,11 @@ void dvb_ca_en50221_release(struct dvb_ca_en50221 *pubca)
    dprintk("%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_running) {
+     ca->exit = 1;
+     mb();
+     dvb_ca_en50221_thread_wakeup(ca);

```

```
+ wait_event_interruptible(ca->thread_queue, ca->thread_running == 0);  
}  
  
for (i = 0; i < ca->slot_count; i++) {  
--  
1.5.0.g53756
```

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
