Subject: Re: [PATCH] dvb_en_50221: Convert to kthread API
Posted by Christoph Hellwig on Fri, 20 Apr 2007 06:37:14 GMT
View Forum Message <> Reply to Message

On Thu, Apr 19, 2007 at 03:34:13PM -0700, Andrew Morton wrote:
> On Thu, 19 Apr 2007 01:59:04 -0600
> "Eric W. Biederman" <ebiederm@xmission.com> wrote:
>
> > 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.
>
> urgh, yes, this is just sad. We should convert this driver fully to
> the kthread API - it will end up much better.
>
> I'll queue this up as a -mm-only thing as a gentle reminder that
> we should do it properly.

Here's an attempted update to the full kthread API + wake_up_process:

Index: linux-2.6/drivers/media/dvb/dvb-core/dvb_ca_en50221.c
===
--- linux-2.6.orig/drivers/media/dvb/dvb-core/dvb_ca_en50221.c 2007-04-20 07:25:07.000000000 +0200
+++ linux-2.6/drivers/media/dvb/dvb-core/dvb_ca_en50221.c 2007-04-20 07:35:54.000000000 +0200
@@ -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"
@@ -140,13 +141,7 @@ struct dvb_ca_private {
  wait_queue_head_t wait_queue;

  /* PID of the monitoring thread */
- pid_t thread_pid;
-
- /* Wait queue used when shutting thread down */
- wait_queue_head_t thread_queue;
-
- /* Flag indicating when thread should exit */
- unsigned int exit:1;
+ struct task_struct *thread;

  /* Flag indicating if the CA device is open */
  unsigned int open:1;
@@ -902,28 +897,10 @@ static void dvb_ca_en50221_thread_wakeup

  ca->wakeup = 1;
  mb();
- wake_up_interruptible(&ca->thread_queue);
+ wake_up_process(ca->thread);
 }

 /**
- * Used by the CA thread to determine if an early wakeup is necessary
- *
- * @param ca CA instance.
- */
-static int dvb_ca_en50221_thread_should_wakeup(struct dvb_ca_private *ca)
-{
- if (ca->wakeup) {
-  ca->wakeup = 0;
-  return 1;
- }
- if (ca->exit)
-  return 1;
-
- return 0;
-}
-
-
-/**
  * Update the delay used by the thread.
  *
  * @param ca CA instance.
@@ -982,7 +959,6 @@ 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 +967,17 @@ 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 (!kthread_should_stop()) {
   /* 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 */
-    break;
-   }
+  while (!ca->wakeup) {
+   set_current_state(TASK_INTERRUPTIBLE);
+   schedule_timeout(ca->delay);
+   if (kthread_should_stop())
+    return 0;
   }
   ca->wakeup = 0;

@@ -1181,10 +1146,6 @@ static int dvb_ca_en50221_thread(void *d
   }
  }

- /* completed */
- ca->thread_pid = 0;
- mb();
```

```
- wake_up_interruptible(&ca->thread_queue);
  return 0;
 }

@@ -1682,9 +1643,6 @@ int dvb_ca_en50221_init(struct dvb_adapt
    goto error;
   }
   init_waitqueue_head(&ca->wait_queue);
- ca->thread_pid = 0;
- init_waitqueue_head(&ca->thread_queue);
- ca->exit = 0;
   ca->open = 0;
   ca->wakeup = 0;
   ca->next_read_slot = 0;
@@ -1710,14 +1668,14 @@ int dvb_ca_en50221_init(struct dvb_adapt
   mb();

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

 error:
@@ -1748,17 +1706,7 @@ void dvb_ca_en50221_release(struct dvb_c
   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);
```

```
-  }
- }
+ kthread_stop(ca->thread);

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

_____

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