
Subject: Re: [PATCH] usbatm: Update to use the kthread api.
Posted by [Cedric Le Goater](#) on Thu, 14 Dec 2006 11:13:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Duncan,

Duncan Sands wrote:

>> Well I just took a quick look through them to be certain
>> and I don't see anything that would. Even inside of the guts of
>> request firmware. So I'm pretty certain that SIGTERM was something
>> originally copied from another kernel_thread implementation and
>> wound up being dead code.

>
> Not at all, it was all written from scratch (so now you know who to
> blame :)). And the signal *is* used, as explained in my reply to
> your original email.

Here's one I have been keeping for a while. Nothing really fancy :
basic replacement of kernel_thread and removal of the start
completion which is now covered in kthread.

Cheers,

C.

Subject: replace kernel_thread() with kthread_run() in usbatm

From: Cedric Le Goater <clg@fr.ibm.com>

This patch replaces the kernel_thread() with kthread_run() since
kernel_thread() is deprecated in drivers/modules.

Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>

Cc: Duncan Sands <duncan.sands@free.fr>

Cc: linux-usb-users@lists.sourceforge.net

Cc: linux-usb-devel@lists.sourceforge.net

drivers/usb/atm/usbatm.c | 26 ++++++++-----
drivers/usb/atm/usbatm.h | 3 +--
2 files changed, 13 insertions(+), 16 deletions(-)

Index: 2.6.19-mm1/drivers/usb/atm/usbatm.c

=====

--- 2.6.19-mm1.orig/drivers/usb/atm/usbatm.c
+++ 2.6.19-mm1/drivers/usb/atm/usbatm.c
@@ -81,6 +81,7 @@
#include <linux/stat.h>

```

#include <linux/timer.h>
#include <linux/wait.h>
#include <linux/kthread.h>

#ifdef VERBOSE_DEBUG
static int usbatm_print_packet(const unsigned char *data, int len);
@@ -999,11 +1000,7 @@ static int usbatm_do_heavy_init(void *ar
    struct usbatm_data *instance = arg;
    int ret;

- daemonize(instance->driver->driver_name);
  allow_signal(SIGTERM);
- instance->thread_pid = current->pid;
-
- complete(&instance->thread_started);

    ret = instance->driver->heavy_init(instance, instance->usb_intf);

@@ -1011,7 +1008,7 @@ static int usbatm_do_heavy_init(void *ar
    ret = usbatm_atm_init(instance);

    mutex_lock(&instance->serialize);
- instance->thread_pid = -1;
+ instance->thread_task = NULL;
    mutex_unlock(&instance->serialize);

    complete_and_exit(&instance->thread_exited, ret);
@@ -1019,15 +1016,17 @@ static int usbatm_do_heavy_init(void *ar

static int usbatm_heavy_init(struct usbatm_data *instance)
{
- int ret = kernel_thread(usbatm_do_heavy_init, instance, CLONE_KERNEL);
+ instance->thread_task = kthread_run(usbatm_do_heavy_init, instance,
+     instance->driver->driver_name);

- if (ret < 0) {
-     usb_err(instance, "%s: failed to create kernel_thread (%d)!\n", __func__, ret);
+ if (IS_ERR(instance->thread_task)) {
+     int ret = PTR_ERR(instance->thread_task);
+     usb_err(instance, "%s: failed to create kthread (%d)!\n",
+         __func__, ret);
+     instance->thread_task = NULL;
    return ret;
}

- wait_for_completion(&instance->thread_started);
-
    return 0;

```

```
}
```

```
@@ -1109,8 +1108,7 @@ int usbatm_usb_probe(struct usb_interfac  
    kref_init(&instance->refcount); /* dropped in usbatm_usb_disconnect */  
    mutex_init(&instance->serialize);
```

```
- instance->thread_pid = -1;  
- init_completion(&instance->thread_started);  
+ instance->thread_task = NULL;  
  init_completion(&instance->thread_exited);
```

```
INIT_LIST_HEAD(&instance->vcc_list);  
@@ -1272,8 +1270,8 @@ void usbatm_usb_disconnect(struct usb_in
```

```
    mutex_lock(&instance->serialize);  
    instance->disconnected = 1;  
- if (instance->thread_pid >= 0)  
-   kill_proc(instance->thread_pid, SIGTERM, 1);  
+ if (instance->thread_task)  
+   send_sig(SIGTERM, instance->thread_task, 1);  
    mutex_unlock(&instance->serialize);
```

```
    wait_for_completion(&instance->thread_exited);
```

```
Index: 2.6.19-mm1/drivers/usb/atm/usbatm.h
```

```
=====
```

```
--- 2.6.19-mm1.orig/drivers/usb/atm/usbatm.h
```

```
+++ 2.6.19-mm1/drivers/usb/atm/usbatm.h
```

```
@@ -176,8 +176,7 @@ struct usbatm_data {  
    int disconnected;
```

```
    /* heavy init */
```

```
- int thread_pid;  
- struct completion thread_started;  
+ struct task_struct *thread_task;  
  struct completion thread_exited;
```

```
    /* ATM device */
```

Containers mailing list

Containers@lists.osdl.org

<https://lists.osdl.org/mailman/listinfo/containers>
