
Subject: Re: Re: [PATCH] bluetooth bnep: Convert to kthread API.
Posted by [Cedric Le Goater](#) on Fri, 20 Apr 2007 12:37:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Cedric Le Goater wrote:

> Andrew Morton wrote:

>> On Thu, 19 Apr 2007 01:58:51 -0600

>> "Eric W. Biederman" <ebiederm@xmission.com> wrote:

>>

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

>>>

>>> This patch starts kbnepd using kthread_run replacing
>>> a combination of kernel_thread and daemonize. Making
>>> the code a little simpler and more maintainable.

>>>

>>>

>> while (!atomic_read(&s->killed)) {

>>

>> ho hum.

>

>

> yes. we need something like :

>

> - while (!atomic_read(&s->killed)) {

> + while (1) {

> + try_to_freeze();

>

> + set_current_state(TASK_INTERRUPTIBLE);

>

> + if (atomic_read(&s->killed))

> + break;

> +

>

> I have an old patch for this driver. I'll refresh it.

>

>>> + task = kthread_run(bnep_session, s, "kbnepd %s", dev->name);

>> It's unusual to have a kernel thread which has a space in its name. That
>> could trip up insufficient-defensive userspace tools.

>

> but we can't just change it, can we ? i could be used by a user space tool
> to check if the thread is running.

here's the refreshed version not taking into account the space in its
kernel thread name.

C.

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

net/bluetooth/bnep/core.c | 15 ++++++-----
1 file changed, 10 insertions(+), 5 deletions(-)

Index: 2.6.21-rc6-mm1/net/bluetooth/bnep/core.c

=====

--- 2.6.21-rc6-mm1.orig/net/bluetooth/bnep/core.c

+++ 2.6.21-rc6-mm1/net/bluetooth/bnep/core.c

@ @ -47,6 +47,7 @ @

#include <linux/netdevice.h>

#include <linux/etherdevice.h>

#include <linux/skbuff.h>

+#include <linux/kthread.h>

#include <asm/unaligned.h>

@ @ -473,16 +474,18 @ @ static int bnep_session(void *arg)

BT_DBG("");

- daemonize("kbnepd %s", dev->name);
set_user_nice(current, -15);

init_waitqueue_entry(&wait, current);
add_wait_queue(sk->sk_sleep, &wait);
- while (!atomic_read(&s->killed)) {
+ while (1) {
try_to_freeze();

set_current_state(TASK_INTERRUPTIBLE);

+ if (atomic_read(&s->killed))
+ break;
+

// RX

while ((skb = skb_dequeue(&sk->sk_receive_queue))) {
skb_orphan(skb);

@ @ -539,6 +542,7 @ @ static struct device *bnep_get_device(st

int bnep_add_connection(struct bnep_connadd_req *req, struct socket *sock)
{

+ struct task_struct *task;

struct net_device *dev;

struct bnep_session *s, *ss;

u8 dst[ETH_ALEN], src[ETH_ALEN];

@ @ -598,9 +602,10 @ @ int bnep_add_connection(struct bnep_conn

```
__bnep_link_session(s);
```

```
- err = kernel_thread(bnep_session, s, CLONE_KERNEL);  
- if (err < 0) {  
- /* Session thread start failed, gotta cleanup. */  
+ task = kthread_run(bnep_session, s, "kbnepd %s", dev->name);  
+ if (IS_ERR(task)) {  
+ /* Session thread start failed, gotta cleanup. */  
+ err = PTR_ERR(task);  
  unregister_netdev(dev);  
  __bnep_unlink_session(s);  
  goto failed;
```
