Subject: [patch -mm] update mq_notify to use a struct pid
Posted by Cedric Le Goater on Fri, 08 Sep 2006 16:39:31 GMT
View Forum Message <> Reply to Message

message queues can signal a process waiting for a message.

this patch replaces the pid_t value with a struct pid to avoid pid wrap
around problems.

Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>
Cc: Eric Biederman <ebiederm@xmission.com>
Cc: Andrew Morton <akpm@osdl.org>
Cc: containers@lists.osdl.org

```
---
 ipc/mqueue.c |   27 +++++++++++++++------------
 1 file changed, 15 insertions(+), 12 deletions(-)

Index: 2.6.18-rc6-mm1/ipc/mqueue.c
===================================================================
--- 2.6.18-rc6-mm1.orig/ipc/mqueue.c
+++ 2.6.18-rc6-mm1/ipc/mqueue.c
@@ -73,7 +73,7 @@ struct mqueue_inode_info {
  struct mq_attr attr;

  struct sigevent notify;
- pid_t notify_owner;
+ struct pid* notify_owner;
  struct user_struct *user; /* user who created, for accounting */
  struct sock *notify_sock;
  struct sk_buff *notify_cookie;
@@ -134,7 +134,7 @@ static struct inode *mqueue_get_inode(st
    INIT_LIST_HEAD(&info->e_wait_q[0].list);
    INIT_LIST_HEAD(&info->e_wait_q[1].list);
    info->messages = NULL;
-   info->notify_owner = 0;
+   info->notify_owner = NULL;
    info->qsize = 0;
    info->user = NULL; /* set when all is ok */
    memset(&info->attr, 0, sizeof(info->attr));
@@ -338,7 +338,7 @@ static ssize_t mqueue_read_file(struct f
    (info->notify_owner &&
     info->notify.sigev_notify == SIGEV_SIGNAL) ?
     info->notify.sigev_signo : 0,
-   info->notify_owner);
+   pid_nr(info->notify_owner));
  spin_unlock(&info->lock);
  buffer[sizeof(buffer)-1] = '\0';
  slen = strlen(buffer)+1;
@@ -363,7 +363,7 @@ static int mqueue_flush_file(struct file
  struct mqueue_inode_info *info = MQUEUE_I(filp->f_dentry->d_inode);

  spin_lock(&info->lock);
- if (current->tgid == info->notify_owner)
+ if (task_tgid(current) == info->notify_owner)
   remove_notification(info);

  spin_unlock(&info->lock);
@@ -518,8 +518,8 @@ static void __do_notify(struct mqueue_in
    sig_i.si_pid = current->tgid;
    sig_i.si_uid = current->uid;

-   kill_proc_info(info->notify.sigev_signo,
-           &sig_i, info->notify_owner);
+   kill_pid_info(info->notify.sigev_signo,
+          &sig_i, info->notify_owner);
    break;
   case SIGEV_THREAD:
    set_cookie(info->notify_cookie, NOTIFY_WOKENUP);
@@ -528,7 +528,8 @@ static void __do_notify(struct mqueue_in
    break;
   }
   /* after notification unregisters process */
-  info->notify_owner = 0;
+  put_pid(info->notify_owner);
+  info->notify_owner = NULL;
  }
  wake_up(&info->wait_q);
 }
@@ -566,12 +567,13 @@ static long prepare_timeout(const struct

 static void remove_notification(struct mqueue_inode_info *info)
 {
- if (info->notify_owner != 0 &&
+ if (info->notify_owner != NULL &&
     info->notify.sigev_notify == SIGEV_THREAD) {
   set_cookie(info->notify_cookie, NOTIFY_REMOVED);
   netlink_sendskb(info->notify_sock, info->notify_cookie, 0);
  }
- info->notify_owner = 0;
+ put_pid(info->notify_owner);
+ info->notify_owner = NULL;
 }

 static int mq_attr_ok(struct mq_attr *attr)
@@ -1062,11 +1064,11 @@ retry:
  ret = 0;
  spin_lock(&info->lock);
  if (u_notification == NULL) {
-  if (info->notify_owner == current->tgid) {
+  if (info->notify_owner == task_tgid(current)) {
    remove_notification(info);
    inode->i_atime = inode->i_ctime = CURRENT_TIME;
   }
- } else if (info->notify_owner != 0) {
+ } else if (info->notify_owner != NULL) {
   ret = -EBUSY;
  } else {
   switch (notification.sigev_notify) {
@@ -1086,7 +1088,8 @@ retry:
    info->notify.sigev_notify = SIGEV_SIGNAL;
    break;
   }
-  info->notify_owner = current->tgid;
+
+  info->notify_owner = get_pid(task_tgid(current));
   inode->i_atime = inode->i_ctime = CURRENT_TIME;
  }
  spin_unlock(&info->lock);
```

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [patch -mm] update mq_notify to use a struct pid
Posted by ebiederm on Sat, 09 Sep 2006 02:39:05 GMT

View Forum Message <> Reply to Message

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

> message queues can signal a process waiting for a message.
>
> this patch replaces the pid_t value with a struct pid to avoid pid wrap
> around problems.
>
> Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>
> Cc: Eric Biederman <ebiederm@xmission.com>
> Cc: Andrew Morton <akpm@osdl.org>
> Cc: containers@lists.osdl.org

Signed-off-by: Eric Biederman <ebiederm@xmission.com>

I was just about to send out this patch in a couple more hours.

So expect the fact we wrote the same code is a good sign :)

Eric

Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [patch -mm] update mq_notify to use a struct pid
Posted by Cedric Le Goater on Mon, 11 Sep 2006 10:17:26 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> Cedric Le Goater <clg@fr.ibm.com> writes:
>
>> message queues can signal a process waiting for a message.
>>
>> this patch replaces the pid_t value with a struct pid to avoid pid wrap
>> around problems.
>>
>> Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>
>> Cc: Eric Biederman <ebiederm@xmission.com>
>> Cc: Andrew Morton <akpm@osdl.org>
>> Cc: containers@lists.osdl.org
>
> Signed-off-by: Eric Biederman <ebiederm@xmission.com>
>
> I was just about to send out this patch in a couple more hours.

Well, you did the same with the usb/devio.c and friends :)

> So expect the fact we wrote the same code is a good sign :)

How does oleg feel about it ? I've seen some long thread on possible race
conditions with put_pid() and solutions with rcu. I didn't quite get all of
it ... it will need another run for me.

On the "pid_t to struct pid*" topic:

* I started smbfs and realized it was useless.

* in the following, the init process is being killed directly using 1. I'm
not sure how useful it would be to use a struct pid. To begin with, may be
they could use a :

kill_init(int signum, int priv)

./arch/mips/sgi-ip32/ip32-reset.c
./arch/powerpc/platforms/iseries/mf.c
./drivers/parisc/power.c
./drivers/char/snsc_event.c
./kernel/sys.c
./kernel/sysctl.c
./drivers/char/nwbutton.c
./drivers/s390/s390mach.c

* some more drivers,
* some more kthread to convert

C.

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [patch -mm] update mq_notify to use a struct pid
Posted by ebiederm on Mon, 11 Sep 2006 11:09:19 GMT

View Forum Message <> Reply to Message

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

> Eric W. Biederman wrote:
>> Cedric Le Goater <clg@fr.ibm.com> writes:
>>
>>> message queues can signal a process waiting for a message.
>>>
>>> this patch replaces the pid_t value with a struct pid to avoid pid wrap
>>> around problems.
>>>
>>> Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>
>>> Cc: Eric Biederman <ebiederm@xmission.com>
>>> Cc: Andrew Morton <akpm@osdl.org>
>>> Cc: containers@lists.osdl.org
>>
>> Signed-off-by: Eric Biederman <ebiederm@xmission.com>
>>
>> I was just about to send out this patch in a couple more hours.
>
> Well, you did the same with the usb/devio.c and friends :)

Good. The you should be familiar enough with it to review my patch and make certain I didn't do anything stupid :)

>> So expect the fact we wrote the same code is a good sign :)
>
> How does oleg feel about it ? I've seen some long thread on possible race
> conditions with put_pid() and solutions with rcu. I didn't quite get all of
> it ... it will need another run for me.

Short. Oleg felt it was a shame that locking was needed to use a
struct pid.

While parsing that I realized my second vt patch that deals with
vt_pid (the pid for console switching) has a subtle race, and
that patch needs to be reworked.

We confused each other. :)

> On the "pid_t to struct pid*" topic:
>
> * I started smbfs and realized it was useless.

Killing the user space part is useless?
I thought that is what I saw happening.

Of course I don't frequently mount smbfs.

> * in the following, the init process is being killed directly using 1. I'm
> not sure how useful it would be to use a struct pid. To begin with, may be
> they could use a :
>
> kill_init(int signum, int priv)

An interesting notion. The other half of them use cad_pid.
Converting that is going to need some sysctl work, so I have been
ignoring it temporarily.

Filling in a struct pid through sysctl is extremely ugly at the
moment, plus cad_pid needs some locking.

> ./arch/mips/sgi-ip32/ip32-reset.c
> ./arch/powerpc/platforms/iseries/mf.c
> ./drivers/parisc/power.c
> ./drivers/char/snsc_event.c
> ./kernel/sys.c
> ./kernel/sysctl.c
> ./drivers/char/nwbutton.c
> ./drivers/s390/s390mach.c
>

> * some more drivers,
> * some more kthread to convert

Ok. Time to exchange some status information, before I roll over and
go back to sleep.

My patch todo list (almost a series file) currently looks like:
> n_r396r
> fs3270-Change-to-use-struct-pid.txt
> smbfs-Make-conn_pid-a-struct-pid.txt
> ncpfs-Use-struct-pid-to-track-the-userspace-watchdog-process.txt
>
> Don-t-use-kill_pg-in-the-sunos-compatibility-code.txt
>
> usbatm-use-kthread-api (I think I have this one)
I did usbatm mostly to figure out why kthread conversions seem
to be so hard, and got lucky this one wasn't too ugly.

> The-dvb_core-needs-to-use-the-kthread-api-not-kernel-threads.txt
> nfs-Note-we-need-to-start-using-the-kthreads-api.txt

dvb-core I have only started looking at.
nfs I noticed it is the svc stuff that matters.

usbatm, dvb-core, and nfs are the 3 kernel_thread users
that also use kill_proc, and thus are high on my immediate hit list.

> pid-Replace-session_of_pgrp-with-pgrp_in_current_session.txt
> pid-Use-struct-pid-for-talking-about-process-groups-in-exit.c.txt
> pid-Replace-is_orphaned_pgrp-with-is_current_pgrp_orphaned.txt
>
> tty-Update-the-tty-layer-to-work-with-struct-pid.txt
I need to ensure I don't have a race with task->signal->tty_old_pgrp.
tty_old_pgrp is a weird notion that I haven't fully wrapped my head
around yet.

> pid-Remove-use-of-old-do_each_task_pid-while_each_task_pid.txt
>
> Rewrite-kill_something_info-so-it-uses-newer-helpers.txt
>
> pid-Remove-now-unused-do_each_task_pid-and-while_each_task_pid.txt
> Remove-the-now-unused-kill_pg-kill_pg_info-and-__kill_pg_info.txt
>
>
> pid-Better-tests-for-same-thread-group-membership.txt
> pid-Cleanup-the-pid-equality-tests.txt

> pid-Track-the-sending-pid-of-a-queued-signal.txt
> proc-Use-pid_nr-in-array.c-so-the-code-is-foobar-safe.txt
>
> sysctl-Implement-get_data-put_data.txt
>
> cad-pid (killing init)

Once the above list is processed none of the old none of the signal
functions that take a pid_t is needed anymore.
i.e. kill_proc, kill_pg, and do_each_task_pid will be removable.

I have at least a first draft of everything on my list except for the
kthread conversions which I just started messing with yesterday.   But
don't worry about beating me to something if you feel you have it
complete. That just means I will have enough of a clue to review your
code :)

Eric

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [patch -mm] update mq_notify to use a struct pid
Posted by Cedric Le Goater on Mon, 11 Sep 2006 14:05:50 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:

>>> I was just about to send out this patch in a couple more hours.
>> Well, you did the same with the usb/devio.c and friends :)
>
> Good. The you should be familiar enough with it to review my patch
> and make certain I didn't do anything stupid :)

well, the least i can try ...

>>> So expect the fact we wrote the same code is a good sign :)
>> How does oleg feel about it ? I've seen some long thread on possible race
>> conditions with put_pid() and solutions with rcu. I didn't quite get all of
>> it ... it will need another run for me.
>
> Short. Oleg felt it was a shame that locking was needed to use a
> struct pid.
>
> While parsing that I realized my second vt patch that deals with

> vt_pid (the pid for console switching) has a subtle race, and
> that patch needs to be reworked.
>
> We confused each other. :)
>
>> On the "pid_t to struct pid*" topic:
>>
>> * I started smbfs and realized it was useless.
>
> Killing the user space part is useless?
> I thought that is what I saw happening.

smb_fill_super() says :

```
 if (warn_count < 5) {
  warn_count++;
  printk(KERN_EMERG "smbfs is deprecated and will be removed in"
    " December, 2006.  Please migrate to cifs\n");
 }
```

So, i guess we should forget about it and spend our time on the cifs
kthread instead.

> Of course I don't frequently mount smbfs.
>
>> * in the following, the init process is being killed directly using 1. I'm
>> not sure how useful it would be to use a struct pid. To begin with, may be
>> they could use a :
>>
>> kill_init(int signum, int priv)
>
> An interesting notion. The other half of them use cad_pid.

yes.

> Converting that is going to need some sysctl work, so I have been
> ignoring it temporarily.
>
> Filling in a struct pid through sysctl is extremely ugly at the
> moment, plus cad_pid needs some locking.

Which distros use /proc/sys/kernel/cad_pid and why ? I can image the need
but i didn't find much on the topic.

> My patch todo list (almost a series file) currently looks like:
>> n_r396r
>> fs3270-Change-to-use-struct-pid.txt

done that. will send to martin for review.

>> smbfs-Make-conn_pid-a-struct-pid.txt

deprecated in december. so we could just forget about it.

>> ncpfs-Use-struct-pid-to-track-the-userspace-watchdog-process.txt
>>
>> Don-t-use-kill_pg-in-the-sunos-compatibility-code.txt
>>
>> usbatm-use-kthread-api (I think I have this one)
> I did usbatm mostly to figure out why kthread conversions seem
> to be so hard, and got lucky this one wasn't too ugly.

argh. i've done also and i just send my second version of the patch to the
maintainer Duncan Sands.

This one might just be useless also because greg kh has a patch in -mm to
enable multithread probing of USB devices.

>> The-dvb_core-needs-to-use-the-kthread-api-not-kernel-threads.txt
>> nfs-Note-we-need-to-start-using-the-kthreads-api.txt
>
> dvb-core I have only started looking at.

suka and i have sent patches to fix :

drivers/media/video/tvaudio.c
drivers/media/video/saa7134/saa7134-tvaudio.c

we are no waiting for the maintainer feedback.

> nfs I noticed it is the svc stuff that matters.
>
> usbatm, dvb-core, and nfs are the 3 kernel_thread users
> that also use kill_proc, and thus are high on my immediate hit list.

nfs is also full of signal_pending() ...

>> pid-Replace-session_of_pgrp-with-pgrp_in_current_session.txt
>> pid-Use-struct-pid-for-talking-about-process-groups-in-exit.c.txt
>> pid-Replace-is_orphaned_pgrp-with-is_current_pgrp_orphaned.txt
>>
>> tty-Update-the-tty-layer-to-work-with-struct-pid.txt
> I need to ensure I don't have a race with task->signal->tty_old_pgrp.
> tty_old_pgrp is a weird notion that I haven't fully wrapped my head
> around yet.
>

>> pid-Remove-use-of-old-do_each_task_pid-while_each_task_pid.txt
>>
>> Rewrite-kill_something_info-so-it-uses-newer-helpers.txt
>>
>> pid-Remove-now-unused-do_each_task_pid-and-while_each_task_pid.txt
>> Remove-the-now-unused-kill_pg-kill_pg_info-and-__kill_pg_info.txt
>>
>>
>> pid-Better-tests-for-same-thread-group-membership.txt
>> pid-Cleanup-the-pid-equality-tests.txt
>> pid-Track-the-sending-pid-of-a-queued-signal.txt

is that about updating the siginfos in collect_signal() to hold the right
pid value depending on the pid namespace they are being received ?

>> proc-Use-pid_nr-in-array.c-so-the-code-is-foobar-safe.txt
>>
>> sysctl-Implement-get_data-put_data.txt
>>
>> cad-pid (killing init)
>
> Once the above list is processed none of the old none of the signal
> functions that take a pid_t is needed anymore.
> i.e. kill_proc, kill_pg, and do_each_task_pid will be removable.
>
> I have at least a first draft of everything on my list except for the
> kthread conversions which I just started messing with yesterday.   But
> don't worry about beating me to something if you feel you have it
> complete. That just means I will have enough of a clue to review your
> code :)

good list ! I look at it in details.

thanks,

C.
____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [patch -mm] update mq_notify to use a struct pid
Posted by Oleg Nesterov on Mon, 11 Sep 2006 15:48:33 GMT
View Forum Message <> Reply to Message

On 09/11, Cedric Le Goater wrote:
>

> Eric W. Biederman wrote:
> > Cedric Le Goater <clg@fr.ibm.com> writes:
> >
> >> message queues can signal a process waiting for a message.
> >>
> >> this patch replaces the pid_t value with a struct pid to avoid pid wrap
> >> around problems.
> >>
> >> Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>
> >> Cc: Eric Biederman <ebiederm@xmission.com>
> >> Cc: Andrew Morton <akpm@osdl.org>
> >> Cc: containers@lists.osdl.org
> >
> > Signed-off-by: Eric Biederman <ebiederm@xmission.com>
> >
> > I was just about to send out this patch in a couple more hours.
>
> Well, you did the same with the usb/devio.c and friends :)
>
> > So expect the fact we wrote the same code is a good sign :)
>
> How does oleg feel about it ? I've seen some long thread on possible race
> conditions with put_pid() and solutions with rcu. I didn't quite get all of
> it ... it will need another run for me.

I assume you are talking about this patch:
 http://marc.theaimsgroup.com/?l=linux-mm-commits&m=115773820415171

I think it's ok, info->notify_owner is always used under info->lock.

This is simple. If, for example, mqueue_read_file() didn't take info->lock,
then we have a problem: pid_nr() may read a freed memory in case when
__do_notify()->put_pid() happens at the same time.

In this context info->notify_owner is a usual refcounted object, no special
attention is needed.

Oleg.

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [patch -mm] update mq_notify to use a struct pid
Posted by ebiederm on Mon, 11 Sep 2006 19:01:18 GMT

Cedric you mentioned a couple of other patches that are in flight.
In the future could you please Cc: the containers list so independent
efforts are less likely to duplicate work, and we are more likely
to review each others patches instead?

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

> Eric W. Biederman wrote:
>
>>>> I was just about to send out this patch in a couple more hours.
>>> Well, you did the same with the usb/devio.c and friends :)
>>
>> Good. The you should be familiar enough with it to review my patch
>> and make certain I didn't do anything stupid :)
>
> well, the least i can try ...
>

>>> * I started smbfs and realized it was useless.
>>
>> Killing the user space part is useless?
>> I thought that is what I saw happening.
>
> smb_fill_super() says :
>
> if (warn_count < 5) {
>   warn_count++;
>   printk(KERN_EMERG "smbfs is deprecated and will be removed in"
>   " December, 2006. Please migrate to cifs\n");
> }
>
> So, i guess we should forget about it and spend our time on the cifs
> kthread instead.

Sure. Although in this instance the changes are simple enough I will probably
send the patch anyway :) That at least explains why you figured it was
useless work.

>> Of course I don't frequently mount smbfs.
>>
>>> * in the following, the init process is being killed directly using 1. I'm
>>> not sure how useful it would be to use a struct pid. To begin with, may be
>>> they could use a :
>>>
>>> kill_init(int signum, int priv)
>>

>> An interesting notion. The other half of them use cad_pid.
>
> yes.
>
>> Converting that is going to need some sysctl work, so I have been
>> ignoring it temporarily.
>>
>> Filling in a struct pid through sysctl is extremely ugly at the
>> moment, plus cad_pid needs some locking.
>
> Which distros use /proc/sys/kernel/cad_pid and why ? I can image the need
> but i didn't find much on the topic.

I'm not at all certain, and I'm not even certain I care. The concept
is there in the code so it needs to be dealt with. Although if I we
extend the cad_pid concept it may make a difference.

>> My patch todo list (almost a series file) currently looks like:
>>> n_r396r
>>> fs3270-Change-to-use-struct-pid.txt
>
> done that. will send to martin for review.

Added to my queue of pending patches to look at review.

>>> ncpfs-Use-struct-pid-to-track-the-userspace-watchdog-process.txt
>>>
>>> Don-t-use-kill_pg-in-the-sunos-compatibility-code.txt
>>>
>>> usbatm-use-kthread-api (I think I have this one)
>> I did usbatm mostly to figure out why kthread conversions seem
>> to be so hard, and got lucky this one wasn't too ugly.
>
> argh. i've done also and i just send my second version of the patch to the
> maintainer Duncan Sands.
>
> This one might just be useless also because greg kh has a patch in -mm to
> enable multithread probing of USB devices.

Added to my queue of pending patches to track down and reivew.

>>> The-dvb_core-needs-to-use-the-kthread-api-not-kernel-threads.txt
>>> nfs-Note-we-need-to-start-using-the-kthreads-api.txt
>>

>> dvb-core I have only started looking at.
>
> suka and i have sent patches to fix :
>
> drivers/media/video/tvaudio.c
> drivers/media/video/saa7134/saa7134-tvaudio.c
>
> we are no waiting for the maintainer feedback.

Ok. I think I saw a little of that.

>> nfs I noticed it is the svc stuff that matters.
>>
>> usbatm, dvb-core, and nfs are the 3 kernel_thread users
>> that also use kill_proc, and thus are high on my immediate hit list.
>
> nfs is also full of signal_pending() ...

Yes, there is a lot to read and understand before I can confidently
do much with nfs.

>>> pid-Better-tests-for-same-thread-group-membership.txt
>>> pid-Cleanup-the-pid-equality-tests.txt
>>> pid-Track-the-sending-pid-of-a-queued-signal.txt
>
> is that about updating the siginfos in collect_signal() to hold the right
> pid value depending on the pid namespace they are being received ?

Yes in send_signal, and in collect signal. To make it work easily I needed
to add a struct pid to struct sigqueue. So in send_signal I generate
the struct pid from the pid_t value and in collect signal I regenerate
the numeric value.

Eric

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [patch -mm] update mq_notify to use a struct pid
Posted by Cedric Le Goater on Mon, 11 Sep 2006 21:53:33 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:

> Cedric you mentioned a couple of other patches that are in flight.
> In the future could you please Cc: the containers list so independent
> efforts are less likely to duplicate work, and we are more likely
> to review each others patches instead?

yes sure, i was relying on the openvz wiki to avoid duplicated efforts on
this topic but i guess email is just the one and only tool for this kind of
development :)

[...]

>>> Converting that is going to need some sysctl work, so I have been
>>> ignoring it temporarily.
>>>
>>> Filling in a struct pid through sysctl is extremely ugly at the
>>> moment, plus cad_pid needs some locking.
>> Which distros use /proc/sys/kernel/cad_pid and why ? I can image the need
>> but i didn't find much on the topic.
>
> I'm not at all certain, and I'm not even certain I care. The concept
> is there in the code so it needs to be dealt with.

OK. It would be nice to make sure this is still in use before trying to
deal with /proc/sys/kernel/cad_pid.

> Although if I we extend the cad_pid concept it may make a difference.

what do you mean by extending cad_pid ? kill_init() ?

[...]

>> is that about updating the siginfos in collect_signal() to hold the right
>> pid value depending on the pid namespace they are being received ?
>
> Yes in send_signal, and in collect signal. To make it work easily I needed
> to add a struct pid to struct sigqueue. So in send_signal I generate
> the struct pid from the pid_t value and in collect signal I regenerate
> the numeric value.

OK. That's what i imagined also but we need a bit more of the pid namespace
to regenerate the numerical value. So, how will you convert this 'struct
pid*' in a pid value using the current pid namespace ?

thinking aloud :

* if the pid namespace of the sending struct pid and current match,
 use nr.
* if they don't,

 if the sending pid namespace is the ancestor of the current pid
 namespace
  use 0
 else
  it's a bug.

struct pid* needs a pid namespace attribute and pid namespace needs to know
its parent.

C.

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [patch -mm] update mq_notify to use a struct pid
Posted by ebiederm on Tue, 12 Sep 2006 01:22:20 GMT
View Forum Message <> Reply to Message

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

> Eric W. Biederman wrote:
>
>> Cedric you mentioned a couple of other patches that are in flight.
>> In the future could you please Cc: the containers list so independent
>> efforts are less likely to duplicate work, and we are more likely
>> to review each others patches instead?
>
> yes sure, i was relying on the openvz wiki to avoid duplicated efforts on
> this topic but i guess email is just the one and only tool for this kind of
> development :)

Sure. Especially when it comes to helping review each others code :)
Not duplicating work is not really my goal, not submitting a patch
after a patch has been reviewed and accepted is.

Plus we need patch review.

Several people working on a patch in parallel if it is difficult
can frequently find a solution that a single person would miss.

>>>> Filling in a struct pid through sysctl is extremely ugly at the
>>>> moment, plus cad_pid needs some locking.
>>> Which distros use /proc/sys/kernel/cad_pid and why ? I can image the need
>>> but i didn't find much on the topic.
>>
>> I'm not at all certain, and I'm not even certain I care. The concept

>> is there in the code so it needs to be dealt with.
>
> OK. It would be nice to make sure this is still in use before trying to
> deal with /proc/sys/kernel/cad_pid.
>
>> Although if I we extend the cad_pid concept it may make a difference.
>
> what do you mean by extending cad_pid ? kill_init() ?

My meaning was every time we are sending a signal to init. It is quite
possible we should be using cad_pid instead.

>>> is that about updating the siginfos in collect_signal() to hold the right
>>> pid value depending on the pid namespace they are being received ?
>>
>> Yes in send_signal, and in collect signal. To make it work easily I needed
>> to add a struct pid to struct sigqueue. So in send_signal I generate
>> the struct pid from the pid_t value and in collect signal I regenerate
>> the numeric value.
>
> OK. That's what i imagined also but we need a bit more of the pid namespace
> to regenerate the numerical value. So, how will you convert this 'struct
> pid*' in a pid value using the current pid namespace ?

By calling pid_nr :) The question I guess is how will pid_nr be implemented.

> thinking aloud :
>
> * if the pid namespace of the sending struct pid and current match,
> use nr.
> * if they don't,
> if the sending pid namespace is the ancestor of the current pid
> namespace
>   use 0
> else
>   it's a bug.
>
> struct pid* needs a pid namespace attribute and pid namespace needs to know
> its parent.

Yes, that sounds correct.

There is also the case that should not come up with signals where
we have a pid from a child namespace, that we should also be able to
compute the pid for.

In essence I intend to have a list of pid_namespace, pid_t pairs connected
to a struct pid that we can look through to find the appropriate pid.

Eric

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [patch -mm] update mq_notify to use a struct pid
Posted by Herbert Poetzl on Tue, 12 Sep 2006 11:05:59 GMT

View Forum Message <> Reply to Message

On Mon, Sep 11, 2006 at 01:01:18PM -0600, Eric W. Biederman wrote:
>
> Cedric you mentioned a couple of other patches that are in flight.
> In the future could you please Cc: the containers list so independent
> efforts are less likely to duplicate work, and we are more likely
> to review each others patches instead?
>
> Cedric Le Goater <clg@fr.ibm.com> writes:
>
> > Eric W. Biederman wrote:
> >
> >>>> I was just about to send out this patch in a couple more hours.
> >>> Well, you did the same with the usb/devio.c and friends :)
> >>
> >> Good. The you should be familiar enough with it to review my patch
> >> and make certain I didn't do anything stupid :)
> >
> > well, the least i can try ...
> >
>
> >>> * I started smbfs and realized it was useless.
> >>
> >> Killing the user space part is useless?
> >> I thought that is what I saw happening.
> >
> > smb_fill_super() says :
> >
> > if (warn_count < 5) {
> >   warn_count++;
> >   printk(KERN_EMERG "smbfs is deprecated and will be removed in"
> >   " December, 2006. Please migrate to cifs\n");
> > }
> >
> > So, i guess we should forget about it and spend our time on the cifs

> > kthread instead.
>
> Sure. Although in this instance the changes are simple enough I will
> probably send the patch anyway :) That at least explains why you
> figured it was useless work.
>
>
> >> Of course I don't frequently mount smbfs.
> >>
> >>> * in the following, the init process is being killed directly
> >>> using 1. I'm not sure how useful it would be to use a struct pid.
> >>> To begin with, may be they could use a :
> >>>
> >>> kill_init(int signum, int priv)
> >>
> >> An interesting notion. The other half of them use cad_pid.
> >
> > yes.
> >
> >> Converting that is going to need some sysctl work, so I have been
> >> ignoring it temporarily.
> >>
> >> Filling in a struct pid through sysctl is extremely ugly at the
> >> moment, plus cad_pid needs some locking.
> >
> > Which distros use /proc/sys/kernel/cad_pid and why ? I can image the
> > need but i didn't find much on the topic.
>
> I'm not at all certain, and I'm not even certain I care. The concept
> is there in the code so it needs to be dealt with. Although if I we
> extend the cad_pid concept it may make a difference.
>
> >> My patch todo list (almost a series file) currently looks like:
> >>
> >>> n_r396r fs3270-Change-to-use-struct-pid.txt
> >
> > done that. will send to martin for review.
>
> Added to my queue of pending patches to look at review.
>
> >>> ncpfs-Use-struct-pid-to-track-the-userspace-watchdog-process.txt
> >>>
> >>> Don-t-use-kill_pg-in-the-sunos-compatibility-code.txt
> >>>
> >>> usbatm-use-kthread-api (I think I have this one)
> >>
> >> I did usbatm mostly to figure out why kthread conversions seem to
> >> be so hard, and got lucky this one wasn't too ugly.

> >
> > argh. i've done also and i just send my second version of the patch
> > to the maintainer Duncan Sands.
> >
> > This one might just be useless also because greg kh has a patch in
> > -mm to enable multithread probing of USB devices.
>
> Added to my queue of pending patches to track down and reivew.
>
>
> >>> The-dvb_core-needs-to-use-the-kthread-api-not-kernel-threads.txt
> >>> nfs-Note-we-need-to-start-using-the-kthreads-api.txt
> >>
> >> dvb-core I have only started looking at.
> >
> > suka and i have sent patches to fix :
> >
> > drivers/media/video/tvaudio.c
> > drivers/media/video/saa7134/saa7134-tvaudio.c
> >
> > we are no waiting for the maintainer feedback.
>
> Ok. I think I saw a little of that.
>
> >> nfs I noticed it is the svc stuff that matters.
> >>
> >> usbatm, dvb-core, and nfs are the 3 kernel_thread users
> >> that also use kill_proc, and thus are high on my immediate hit list.

> > nfs is also full of signal_pending() ...

> Yes, there is a lot to read and understand before I can confidently
> do much with nfs.

I already did a lot of adjustments to the nfs system, and
I poked aound in dvb-core before, so I will take a look
at this in the next few days, at least the switch to the
kthread api should not be a big deal ...

HTH,
Herbert

> >>> pid-Better-tests-for-same-thread-group-membership.txt
> >>> pid-Cleanup-the-pid-equality-tests.txt
> >>> pid-Track-the-sending-pid-of-a-queued-signal.txt
> >
> > is that about updating the siginfos in collect_signal() to hold the right
> > pid value depending on the pid namespace they are being received ?

>
> Yes in send_signal, and in collect signal. To make it work easily I
> needed to add a struct pid to struct sigqueue. So in send_signal I
> generate the struct pid from the pid_t value and in collect signal I
> regenerate the numeric value.
>
>
> Eric
>
> ____
> Containers mailing list
> Containers@lists.osdl.org
> https://lists.osdl.org/mailman/listinfo/containers

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [patch -mm] update mq_notify to use a struct pid
Posted by ebiederm on Tue, 12 Sep 2006 15:14:02 GMT

View Forum Message <> Reply to Message

Herbert Poetzl <herbert@13thfloor.at> writes:

> I already did a lot of adjustments to the nfs system, and
> I poked aound in dvb-core before, so I will take a look
> at this in the next few days, at least the switch to the
> kthread api should not be a big deal ...

Ok. If you can get this it would be great.

To some extent the last holdouts on the kernel_thread api seem to
be the ones that are not trivial to convert :(

Eric

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [patch -mm] update mq_notify to use a struct pid
Posted by Cedric Le Goater on Tue, 12 Sep 2006 15:37:28 GMT

View Forum Message <> Reply to Message

Eric W. Biederman wrote:

[...]

> There is also the case that should not come up with signals where
> we have a pid from a child namespace, that we should also be able to
> compute the pid for.

I don't understand how a signal can come from a child pid namespace ?

> In essence I intend to have a list of pid_namespace, pid_t pairs connected
> to a struct pid that we can look through to find the appropriate pid.

yes, that's the purpose of pid_nr() I guess.

This list would contain in nearly all cases a single pair (current pid
namespace, pid value). It will contain 2 pairs for a task that has unshared
its pid namespace : a pair for the current pid namespace, that needs to
allocated when unshare() is called, and one pair for the ancestor pid
namespace which is already allocated.

Do you see more ?

C.

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [patch -mm] update mq_notify to use a struct pid
Posted by ebiederm on Tue, 12 Sep 2006 16:03:32 GMT

View Forum Message <> Reply to Message

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

> Eric W. Biederman wrote:
>
> [...]
>
>> There is also the case that should not come up with signals where
>> we have a pid from a child namespace, that we should also be able to
>> compute the pid for.
>
> I don't understand how a signal can come from a child pid namespace ?

SIG_CHLD is the only case where I think we will be sending a signal
from the child pid namespace.

Reading pids from the status files in /proc, from a parent namespace,
is another example where we need to deal with the pid of children.

>> In essence I intend to have a list of pid_namespace, pid_t pairs connected
>> to a struct pid that we can look through to find the appropriate pid.
>
> yes, that's the purpose of pid_nr() I guess.
>
> This list would contain in nearly all cases a single pair (current pid
> namespace, pid value). It will contain 2 pairs for a task that has unshared
> its pid namespace : a pair for the current pid namespace, that needs to
> allocated when unshare() is called, and one pair for the ancestor pid
> namespace which is already allocated.
>
> Do you see more ?

I don't see the list getting longer until we get into a nested pid namespaces.

As long as the interface is well defined for the container in a container
case I don't mind having additional restrictions.

I will note that you can get some extremely weird interactions if
you do things like open a file descriptor in the parent pid namespace.
Fork two children each child in a different pid_namespaces.
fcntl(F_SETOWN) is called in one child, and fcntl(F_GETOWN) is called
in the other child.

So we can't just call BUG_ON, if we have can't find the namespace.
But returning 0 from pid_nr should be fine.

Eric

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: [PATCH/RFC] kthread API conversion for dvb_frontend and av7110
Posted by Herbert Poetzl on Thu, 14 Sep 2006 20:01:03 GMT

View Forum Message <> Reply to Message

Okay, as I promised, I had a first shot at the
dvb kernel_thread to kthread API port, and here
is the result, which is running fine here since
yesterday, including module load/unload and
software suspend (which doesn't work as expected
with or without this patch :), I didn't convert
the dvb_ca_en50221 as I do not have such an

interface, but if the conversion process is fine
with the v4l-dvb maintainers, it should not be
a problem to send a patch for that too ...

best,
Herbert

Signed-off-by: Herbert Poetzl <herbert@13thfloor.at>

```
diff -NurpP linux-2.6.18-rc6/drivers/media/dvb/dvb-core/dvb_frontend.c
linux-2.6.18-rc6-kthread.v02.3/drivers/media/dvb/dvb-core/dvb_frontend.c
--- linux-2.6.18-rc6/drivers/media/dvb/dvb-core/dvb_frontend.c 2006-09-12 18:16:12 +0200
+++ linux-2.6.18-rc6-kthread.v02.3/drivers/media/dvb/dvb-core/dvb_frontend.c 2006-09-14
21:23:37 +0200
@@ -36,6 +36,7 @@
 #include <linux/list.h>
 #include <linux/suspend.h>
 #include <linux/jiffies.h>
+#include <linux/kthread.h>
 #include <asm/processor.h>

 #include "dvb_frontend.h"
@@ -100,7 +101,7 @@ struct dvb_frontend_private {
  struct semaphore sem;
  struct list_head list_head;
  wait_queue_head_t wait_queue;
- pid_t thread_pid;
+ struct task_struct *thread;
  unsigned long release_jiffies;
  unsigned int exit;
  unsigned int wakeup;
@@ -508,19 +509,11 @@ static int dvb_frontend_thread(void *dat
  struct dvb_frontend *fe = data;
  struct dvb_frontend_private *fepriv = fe->frontend_priv;
  unsigned long timeout;
- char name [15];
  fe_status_t s;
  struct dvb_frontend_parameters *params;

  dprintk("%s\n", __FUNCTION__);

- snprintf (name, sizeof(name), "kdvb-fe-%i", fe->dvb->num);
-
- lock_kernel();
- daemonize(name);
- sigfillset(&current->blocked);
- unlock_kernel();
-
```

```
  fepriv->check_wrapped = 0;
  fepriv->quality = 0;
  fepriv->delay = 3*HZ;
@@ -534,14 +527,16 @@ static int dvb_frontend_thread(void *dat
   up(&fepriv->sem);     /* is locked when we enter the thread... */

   timeout = wait_event_interruptible_timeout(fepriv->wait_queue,
-          dvb_frontend_should_wakeup(fe),
-          fepriv->delay);
-  if (0 != dvb_frontend_is_exiting(fe)) {
+   dvb_frontend_should_wakeup(fe) || kthread_should_stop(),
+   fepriv->delay);
+
+  if (kthread_should_stop() || dvb_frontend_is_exiting(fe)) {
    /* got signal or quitting */
    break;
   }

-  try_to_freeze();
+  if (try_to_freeze())
+   continue;

   if (down_interruptible(&fepriv->sem))
    break;
@@ -591,7 +586,7 @@ static int dvb_frontend_thread(void *dat
    fe->ops.sleep(fe);
  }

- fepriv->thread_pid = 0;
+ fepriv->thread = NULL;
  mb();

  dvb_frontend_wakeup(fe);
@@ -600,7 +595,6 @@ static int dvb_frontend_thread(void *dat

 static void dvb_frontend_stop(struct dvb_frontend *fe)
 {
- unsigned long ret;
  struct dvb_frontend_private *fepriv = fe->frontend_priv;

  dprintk ("%s\n", __FUNCTION__);
@@ -608,33 +602,17 @@ static void dvb_frontend_stop(struct dvb
  fepriv->exit = 1;
  mb();

- if (!fepriv->thread_pid)
-  return;
-
```

```
- /* check if the thread is really alive */
- if (kill_proc(fepriv->thread_pid, 0, 1) == -ESRCH) {
-  printk("dvb_frontend_stop: thread PID %d already died\n",
-    fepriv->thread_pid);
-  /* make sure the mutex was not held by the thread */
-  init_MUTEX (&fepriv->sem);
+ if (!fepriv->thread)
   return;
- }
-
- /* wake up the frontend thread, so it notices that fe->exit == 1 */
- dvb_frontend_wakeup(fe);

- /* wait until the frontend thread has exited */
- ret = wait_event_interruptible(fepriv->wait_queue,0 == fepriv->thread_pid);
- if (-ERESTARTSYS != ret) {
-  fepriv->state = FESTATE_IDLE;
-  return;
- }
+ kthread_stop(fepriv->thread);
+ init_MUTEX (&fepriv->sem);
  fepriv->state = FESTATE_IDLE;

  /* paranoia check in case a signal arrived */
- if (fepriv->thread_pid)
-  printk("dvb_frontend_stop: warning: thread PID %d won't exit\n",
-    fepriv->thread_pid);
+ if (fepriv->thread)
+  printk("dvb_frontend_stop: warning: thread %p won't exit\n",
+    fepriv->thread);
 }

 s32 timeval_usec_diff(struct timeval lasttime, struct timeval curtime)
@@ -684,10 +662,11 @@ static int dvb_frontend_start(struct dvb
 {
  int ret;
  struct dvb_frontend_private *fepriv = fe->frontend_priv;
+ struct task_struct *fe_thread;

  dprintk ("%s\n", __FUNCTION__);

- if (fepriv->thread_pid) {
+ if (fepriv->thread) {
   if (!fepriv->exit)
    return 0;
   else
@@ -701,18 +680,18 @@ static int dvb_frontend_start(struct dvb
  fepriv->state = FESTATE_IDLE;
  fepriv->exit = 0;
- fepriv->thread_pid = 0;
+ fepriv->thread = NULL;
  mb();

- ret = kernel_thread (dvb_frontend_thread, fe, 0);
-
- if (ret < 0) {
-  printk("dvb_frontend_start: failed to start kernel_thread (%d)\n", ret);
+ fe_thread = kthread_run(dvb_frontend_thread, fe,
+  "kdvb-fe-%i", fe->dvb->num);
+ if (IS_ERR(fe_thread)) {
+  ret = PTR_ERR(fe_thread);
+  printk("dvb_frontend_start: failed to start kthread (%d)\n", ret);
   up(&fepriv->sem);
   return ret;
  }
- fepriv->thread_pid = ret;
-
+ fepriv->thread = fe_thread;
  return 0;
 }

diff -NurpP linux-2.6.18-rc6/drivers/media/dvb/ttpci/av7110.c
linux-2.6.18-rc6-kthread.v02.3/drivers/media/dvb/ttpci/av7110.c
--- linux-2.6.18-rc6/drivers/media/dvb/ttpci/av7110.c 2006-09-12 18:16:13 +0200
+++ linux-2.6.18-rc6-kthread.v02.3/drivers/media/dvb/ttpci/av7110.c 2006-09-14 21:21:03 +0200
@@ -51,6 +51,7 @@
 #include <linux/firmware.h>
 #include <linux/crc32.h>
 #include <linux/i2c.h>
+#include <linux/kthread.h>

 #include <asm/system.h>

@@ -223,11 +224,10 @@ static void recover_arm(struct av7110 *a

 static void av7110_arm_sync(struct av7110 *av7110)
 {
- av7110->arm_rmmod = 1;
- wake_up_interruptible(&av7110->arm_wait);
+ if (av7110->arm_thread)
+  kthread_stop(av7110->arm_thread);

- while (av7110->arm_thread)
-  msleep(1);
+ av7110->arm_thread = NULL;
```

```
 }

 static int arm_thread(void *data)
@@ -238,17 +238,11 @@ static int arm_thread(void *data)

  dprintk(4, "%p\n",av7110);

- lock_kernel();
- daemonize("arm_mon");
- sigfillset(&current->blocked);
- unlock_kernel();
-
- av7110->arm_thread = current;
-
  for (;;) {
   timeout = wait_event_interruptible_timeout(av7110->arm_wait,
-          av7110->arm_rmmod, 5 * HZ);
-  if (-ERESTARTSYS == timeout || av7110->arm_rmmod) {
+   kthread_should_stop(), 5 * HZ);
+
+  if (-ERESTARTSYS == timeout || kthread_should_stop()) {
    /* got signal or told to quit*/
    break;
   }
@@ -276,7 +270,6 @@ static int arm_thread(void *data)
   av7110->arm_errors = 0;
  }

- av7110->arm_thread = NULL;
  return 0;
 }

@@ -2334,6 +2327,7 @@ static int __devinit av7110_attach(struc
  const int length = TS_WIDTH * TS_HEIGHT;
  struct pci_dev *pdev = dev->pci;
  struct av7110 *av7110;
+ struct task_struct *thread;
  int ret, count = 0;

  dprintk(4, "dev: %p\n", dev);
@@ -2618,9 +2612,12 @@ static int __devinit av7110_attach(struc
   printk ("dvb-ttpci: Warning, firmware version 0x%04x is too old. "
    "System might be unstable!\n", FW_VERSION(av7110->arm_app));

- ret = kernel_thread(arm_thread, (void *) av7110, 0);
- if (ret < 0)
+ thread = kthread_run(arm_thread, (void *) av7110, "arm_mon");
+ if (IS_ERR(thread)) {
```

```
+  ret = PTR_ERR(thread);
   goto err_stop_arm_9;
+ }
+ av7110->arm_thread = thread;

  /* set initial volume in mixer struct */
  av7110->mixer.volume_left  = volume;
diff -NurpP linux-2.6.18-rc6/drivers/media/dvb/ttpci/av7110.h
linux-2.6.18-rc6-kthread.v02.3/drivers/media/dvb/ttpci/av7110.h
--- linux-2.6.18-rc6/drivers/media/dvb/ttpci/av7110.h 2006-09-12 18:16:13 +0200
+++ linux-2.6.18-rc6-kthread.v02.3/drivers/media/dvb/ttpci/av7110.h 2006-09-14 21:21:03 +0200
@@ -205,7 +205,6 @@ struct av7110 {
  struct task_struct *arm_thread;
  wait_queue_head_t   arm_wait;
  u16      arm_loops;
- int      arm_rmmod;

  void     *debi_virt;
  dma_addr_t     debi_bus;
```

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH/RFC] kthread API conversion for dvb_frontend and av7110
Posted by Cedric Le Goater on Thu, 14 Sep 2006 21:07:49 GMT

View Forum Message <> Reply to Message

Herbert Poetzl wrote:
> Okay, as I promised, I had a first shot at the
> dvb kernel_thread to kthread API port, and here
> is the result, which is running fine here since
> yesterday, including module load/unload and
> software suspend (which doesn't work as expected
> with or without this patch :),

So you have such an hardware ?

[...]

```
> @@ -600,7 +595,6 @@ static int dvb_frontend_thread(void *dat
>
>  static void dvb_frontend_stop(struct dvb_frontend *fe)
>  {
> - unsigned long ret;
>   struct dvb_frontend_private *fepriv = fe->frontend_priv;
>
```

```
>   dprintk ("%s\n", __FUNCTION__);
> @@ -608,33 +602,17 @@ static void dvb_frontend_stop(struct dvb
>   fepriv->exit = 1;
```

do we still need the ->exit flag now that we are using kthread_stop() ?
same question for ->wakeup ?

```
>   mb();
>
> - if (!fepriv->thread_pid)
> -  return;
> -
> - /* check if the thread is really alive */
> - if (kill_proc(fepriv->thread_pid, 0, 1) == -ESRCH) {
> -  printk("dvb_frontend_stop: thread PID %d already died\n",
> -    fepriv->thread_pid);
> -  /* make sure the mutex was not held by the thread */
> -  init_MUTEX (&fepriv->sem);
> + if (!fepriv->thread)
>    return;
> - }
> -
> - /* wake up the frontend thread, so it notices that fe->exit == 1 */
> - dvb_frontend_wakeup(fe);
>
> - /* wait until the frontend thread has exited */
> - ret = wait_event_interruptible(fepriv->wait_queue,0 == fepriv->thread_pid);
> - if (-ERESTARTSYS != ret) {
> -  fepriv->state = FESTATE_IDLE;
> -  return;
> - }
> + kthread_stop(fepriv->thread);
> + init_MUTEX (&fepriv->sem);
```

the use of the semaphore to synchronise the thread is complex. It will
require extra care to avoid deadlocks.

```
>   fepriv->state = FESTATE_IDLE;
>
>   /* paranoia check in case a signal arrived */
> - if (fepriv->thread_pid)
> -  printk("dvb_frontend_stop: warning: thread PID %d won't exit\n",
> -    fepriv->thread_pid);
> + if (fepriv->thread)
> +  printk("dvb_frontend_stop: warning: thread %p won't exit\n",
> +    fepriv->thread);
```

kthread_stop uses a completion already. so the above is real paranoia :)

```
>  }
>
>  s32 timeval_usec_diff(struct timeval lasttime, struct timeval curtime)
> @@ -684,10 +662,11 @@ static int dvb_frontend_start(struct dvb
>  {
>   int ret;
>   struct dvb_frontend_private *fepriv = fe->frontend_priv;
> + struct task_struct *fe_thread;
>
>   dprintk ("%s\n", __FUNCTION__);
>
> - if (fepriv->thread_pid) {
> + if (fepriv->thread) {
>    if (!fepriv->exit)
>     return 0;
>    else
> @@ -701,18 +680,18 @@ static int dvb_frontend_start(struct dvb
>
>   fepriv->state = FESTATE_IDLE;
>   fepriv->exit = 0;
> - fepriv->thread_pid = 0;
> + fepriv->thread = NULL;
>   mb();
>
> - ret = kernel_thread (dvb_frontend_thread, fe, 0);
> -
> - if (ret < 0) {
> -  printk("dvb_frontend_start: failed to start kernel_thread (%d)\n", ret);
> + fe_thread = kthread_run(dvb_frontend_thread, fe,
> +  "kdvb-fe-%i", fe->dvb->num);
> + if (IS_ERR(fe_thread)) {
> +  ret = PTR_ERR(fe_thread);
```

ret could be local.

[...]

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH/RFC] kthread API conversion for dvb_frontend and av7110
Posted by Herbert Poetzl on Thu, 14 Sep 2006 22:10:24 GMT

On Thu, Sep 14, 2006 at 11:07:49PM +0200, Cedric Le Goater wrote:
> Herbert Poetzl wrote:
> > Okay, as I promised, I had a first shot at the
> > dvb kernel_thread to kthread API port, and here
> > is the result, which is running fine here since
> > yesterday, including module load/unload and
> > software suspend (which doesn't work as expected
> > with or without this patch :),
>
> So you have such an hardware ?

yes, I do .. that's how I tested it :)

```
> [ ...  ]
>
> > @@ -600,7 +595,6 @@ static int dvb_frontend_thread(void *dat
> >
> >  static void dvb_frontend_stop(struct dvb_frontend *fe)
> >  {
> > - unsigned long ret;
> >   struct dvb_frontend_private *fepriv = fe->frontend_priv;
> >
> >   dprintk ("%s\n", __FUNCTION__);
> > @@ -608,33 +602,17 @@ static void dvb_frontend_stop(struct dvb
> >   fepriv->exit = 1;
>
> do we still need the ->exit flag now that we are using kthread_stop() ?
> same question for ->wakeup ?
```

probably not, but I didn't want to change too
much on the first try, especially I'd appreciate
some feedback to this from the maintainer(s)

```
> >   mb();
> >
> > - if (!fepriv->thread_pid)
> > -  return;
> > -
> > - /* check if the thread is really alive */
> > - if (kill_proc(fepriv->thread_pid, 0, 1) == -ESRCH) {
> > -  printk("dvb_frontend_stop: thread PID %d already died\n",
> > -    fepriv->thread_pid);
> > -  /* make sure the mutex was not held by the thread */
> > -  init_MUTEX (&fepriv->sem);
> > + if (!fepriv->thread)
> >    return;
> > - }
> > -
> > - /* wake up the frontend thread, so it notices that fe->exit == 1 */
> > - dvb_frontend_wakeup(fe);
> >
> > - /* wait until the frontend thread has exited */
> > - ret = wait_event_interruptible(fepriv->wait_queue,0 == fepriv->thread_pid);
> > - if (-ERESTARTSYS != ret) {
> > -  fepriv->state = FESTATE_IDLE;
> > -  return;
> > - }
> > + kthread_stop(fepriv->thread);
> > + init_MUTEX (&fepriv->sem);
>
> the use of the semaphore to synchronise the thread is complex. It will
> require extra care to avoid deadlocks.
```

well, it 'works' quite fine for now, but yeah, I
thought about completely removing the additional
synchronization and 'jsut' go with the kthread
one, if that is sufficient ...

```
> >   fepriv->state = FESTATE_IDLE;
> >
> >   /* paranoia check in case a signal arrived */
> > - if (fepriv->thread_pid)
> > -  printk("dvb_frontend_stop: warning: thread PID %d won't exit\n",
> > -    fepriv->thread_pid);
> > + if (fepriv->thread)
> > +  printk("dvb_frontend_stop: warning: thread %p won't exit\n",
> > +    fepriv->thread);
>
> kthread_stop uses a completion already. so the above is real paranoia :)
```

again, I think this will go away soon :)

```
> >  }
> >
> >  s32 timeval_usec_diff(struct timeval lasttime, struct timeval curtime)
> > @@ -684,10 +662,11 @@ static int dvb_frontend_start(struct dvb
> >  {
> >   int ret;
> >   struct dvb_frontend_private *fepriv = fe->frontend_priv;
> > + struct task_struct *fe_thread;
> >
> >   dprintk ("%s\n", __FUNCTION__);
> >
> > - if (fepriv->thread_pid) {
> > + if (fepriv->thread) {
```

```
> >     if (!fepriv->exit)
> >      return 0;
> >     else
> > @@ -701,18 +680,18 @@ static int dvb_frontend_start(struct dvb
> >
> >    fepriv->state = FESTATE_IDLE;
> >    fepriv->exit = 0;
> > - fepriv->thread_pid = 0;
> > + fepriv->thread = NULL;
> >    mb();
> >
> > - ret = kernel_thread (dvb_frontend_thread, fe, 0);
> > -
> > - if (ret < 0) {
> > -  printk("dvb_frontend_start: failed to start kernel_thread (%d)\n", ret);
> > + fe_thread = kthread_run(dvb_frontend_thread, fe,
> > +  "kdvb-fe-%i", fe->dvb->num);
> > + if (IS_ERR(fe_thread)) {
> > +  ret = PTR_ERR(fe_thread);
>
> ret could be local.
```

correct, will fix that up in the next round

thanks for the feedback,
Herbert

```
> [ ... ]
>
> _______________________________________________
> Containers mailing list
> Containers@lists.osdl.org
> https://lists.osdl.org/mailman/listinfo/containers
```

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH/RFC] kthread API conversion for dvb_frontend and av7110
Posted by Andrew de Quincey on Fri, 17 Nov 2006 01:50:01 GMT

View Forum Message <> Reply to Message

[snip]

```
> correct, will fix that up in the next round
>
> thanks for the feedback,
```

> Herbert

Hi - the conversion looks good to me.. I can't really offer any more
constructive suggestions beyond what Cedric has already said.

Theres another thread in dvb_ca_en50221.c that could be converted as well
though, hint hint ;)

Apologies for the delay in this reply - I've been hibernating for a bit.

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH/RFC] kthread API conversion for dvb_frontend and av7110
Posted by ebiederm on Tue, 12 Dec 2006 22:58:16 GMT

View Forum Message <> Reply to Message

Andrew de Quincey <adq_dvb@lidskialf.net> writes:

> [snip]
>
>> correct, will fix that up in the next round
>>
>> thanks for the feedback,
>> Herbert
>
> Hi - the conversion looks good to me.. I can't really offer any more
> constructive suggestions beyond what Cedric has already said.
>
> Theres another thread in dvb_ca_en50221.c that could be converted as well
> though, hint hint ;)
>
> Apologies for the delay in this reply - I've been hibernating for a bit.

Guys where are we at on this conversion?

Eric

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH/RFC] kthread API conversion for dvb_frontend and av7110
Posted by Herbert Poetzl on Tue, 12 Dec 2006 23:13:15 GMT

View Forum Message <> Reply to Message

On Tue, Dec 12, 2006 at 03:58:16PM -0700, Eric W. Biederman wrote:
> Andrew de Quincey <adq_dvb@lidskialf.net> writes:
>
> > [snip]
> >
> >> correct, will fix that up in the next round
> >>
> >> thanks for the feedback,
> >> Herbert
> >
> > Hi - the conversion looks good to me.. I can't really offer any more
> > constructive suggestions beyond what Cedric has already said.
> >
> > Theres another thread in dvb_ca_en50221.c that could be converted as well
> > though, hint hint ;)
> >
> > Apologies for the delay in this reply - I've been hibernating for a bit.
>
> Guys where are we at on this conversion?

I can take a look at it in the next few days, but
I have no hardware to test that, so it would be
good to get in contact with somebody who does

best,
Herbert

> Eric

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: Re: [PATCH/RFC] kthread API conversion for dvb_frontend and av7110
Posted by Cedric Le Goater on Wed, 13 Dec 2006 15:55:12 GMT
View Forum Message <> Reply to Message

Herbert Poetzl wrote:
> On Tue, Dec 12, 2006 at 03:58:16PM -0700, Eric W. Biederman wrote:
>> Andrew de Quincey <adq_dvb@lidskialf.net> writes:
>>
>>> [snip]
>>>
>>>> correct, will fix that up in the next round

>>>>
>>>> thanks for the feedback,
>>>> Herbert
>>> Hi - the conversion looks good to me.. I can't really offer any more
>>> constructive suggestions beyond what Cedric has already said.
>>>
>>> Theres another thread in dvb_ca_en50221.c that could be converted as well
>>> though, hint hint ;)
>>>
>>> Apologies for the delay in this reply - I've been hibernating for a bit.
>> Guys where are we at on this conversion?

I said I would but I didn't have much time yet?

however, I didn't see any big issue. no semaphore like in dvb_frontend.c.
just some kill(0) that looks harmless.

> I can take a look at it in the next few days, but
> I have no hardware to test that, so it would be
> good to get in contact with somebody who does

neither do I.

below is an updated version of herbert's dvb_frontend.c patch for 2.6.19-mm1

thanks,

C.

From: Herbert Poetzl <herbert@13thfloor.at>

Okay, as I promised, I had a first shot at the
dvb kernel_thread to kthread API port, and here
is the result, which is running fine here since
yesterday, including module load/unload and
software suspend (which doesn't work as expected
with or without this patch :), I didn't convert
the dvb_ca_en50221 as I do not have such an
interface, but if the conversion process is fine
with the v4l-dvb maintainers, it should not be
a problem to send a patch for that too ...

best,
Herbert

Signed-off-by: Herbert Poetzl <herbert@13thfloor.at>

```
 drivers/media/dvb/dvb-core/dvb_frontend.c |   69 ++++++++++--------------------
 drivers/media/dvb/ttpci/av7110.c          |   29 +++++-------
 drivers/media/dvb/ttpci/av7110.h          |    1
 3 files changed, 37 insertions(+), 62 deletions(-)

Index: 2.6.19-mm1/drivers/media/dvb/dvb-core/dvb_frontend.c
===================================================================
--- 2.6.19-mm1.orig/drivers/media/dvb/dvb-core/dvb_frontend.c
+++ 2.6.19-mm1/drivers/media/dvb/dvb-core/dvb_frontend.c
@@ -36,6 +36,7 @@
 #include <linux/list.h>
 #include <linux/freezer.h>
 #include <linux/jiffies.h>
+#include <linux/kthread.h>
 #include <asm/processor.h>

 #include "dvb_frontend.h"
@@ -100,7 +101,7 @@ struct dvb_frontend_private {
 	struct semaphore sem;
 	struct list_head list_head;
 	wait_queue_head_t wait_queue;
-	pid_t thread_pid;
+	struct task_struct *thread;
 	unsigned long release_jiffies;
 	unsigned int exit;
 	unsigned int wakeup;
@@ -508,19 +509,11 @@ static int dvb_frontend_thread(void *dat
 	struct dvb_frontend *fe = data;
 	struct dvb_frontend_private *fepriv = fe->frontend_priv;
 	unsigned long timeout;
-	char name [15];
 	fe_status_t s;
 	struct dvb_frontend_parameters *params;

 	dprintk("%s\n", __FUNCTION__);

-	snprintf (name, sizeof(name), "kdvb-fe-%i", fe->dvb->num);
-
-	lock_kernel();
-	daemonize(name);
-	sigfillset(&current->blocked);
-	unlock_kernel();
-
 	fepriv->check_wrapped = 0;
 	fepriv->quality = 0;
 	fepriv->delay = 3*HZ;
@@ -534,14 +527,16 @@ static int dvb_frontend_thread(void *dat
 		up(&fepriv->sem);	    /* is locked when we enter the thread... */
   timeout = wait_event_interruptible_timeout(fepriv->wait_queue,
-          dvb_frontend_should_wakeup(fe),
-          fepriv->delay);
-  if (0 != dvb_frontend_is_exiting(fe)) {
+   dvb_frontend_should_wakeup(fe) || kthread_should_stop(),
+   fepriv->delay);
+
+  if (kthread_should_stop() || dvb_frontend_is_exiting(fe)) {
    /* got signal or quitting */
    break;
   }

-  try_to_freeze();
+  if (try_to_freeze())
+   continue;

   if (down_interruptible(&fepriv->sem))
    break;
@@ -591,7 +586,7 @@ static int dvb_frontend_thread(void *dat
    fe->ops.sleep(fe);
  }

- fepriv->thread_pid = 0;
+ fepriv->thread = NULL;
  mb();

  dvb_frontend_wakeup(fe);
@@ -600,7 +595,6 @@ static int dvb_frontend_thread(void *dat

 static void dvb_frontend_stop(struct dvb_frontend *fe)
 {
- unsigned long ret;
  struct dvb_frontend_private *fepriv = fe->frontend_priv;

  dprintk ("%s\n", __FUNCTION__);
@@ -608,33 +602,17 @@ static void dvb_frontend_stop(struct dvb
  fepriv->exit = 1;
  mb();

- if (!fepriv->thread_pid)
+ if (!fepriv->thread)
   return;

- /* check if the thread is really alive */
- if (kill_proc(fepriv->thread_pid, 0, 1) == -ESRCH) {
-  printk("dvb_frontend_stop: thread PID %d already died\n",
-    fepriv->thread_pid);
-  /* make sure the mutex was not held by the thread */
-  init_MUTEX (&fepriv->sem);
-  return;
- }
-
- /* wake up the frontend thread, so it notices that fe->exit == 1 */
- dvb_frontend_wakeup(fe);
-
- /* wait until the frontend thread has exited */
- ret = wait_event_interruptible(fepriv->wait_queue,0 == fepriv->thread_pid);
- if (-ERESTARTSYS != ret) {
-  fepriv->state = FESTATE_IDLE;
-  return;
- }
+ kthread_stop(fepriv->thread);
+ init_MUTEX (&fepriv->sem);
  fepriv->state = FESTATE_IDLE;

  /* paranoia check in case a signal arrived */
- if (fepriv->thread_pid)
-  printk("dvb_frontend_stop: warning: thread PID %d won't exit\n",
-    fepriv->thread_pid);
+ if (fepriv->thread)
+  printk("dvb_frontend_stop: warning: thread %p won't exit\n",
+    fepriv->thread);
 }

 s32 timeval_usec_diff(struct timeval lasttime, struct timeval curtime)
@@ -684,10 +662,11 @@ static int dvb_frontend_start(struct dvb
 {
  int ret;
  struct dvb_frontend_private *fepriv = fe->frontend_priv;
+ struct task_struct *fe_thread;

  dprintk ("%s\n", __FUNCTION__);

- if (fepriv->thread_pid) {
+ if (fepriv->thread) {
   if (!fepriv->exit)
    return 0;
   else
@@ -701,18 +680,18 @@ static int dvb_frontend_start(struct dvb

  fepriv->state = FESTATE_IDLE;
  fepriv->exit = 0;
- fepriv->thread_pid = 0;
+ fepriv->thread = NULL;
  mb();
```

```
- ret = kernel_thread (dvb_frontend_thread, fe, 0);
-
- if (ret < 0) {
-  printk("dvb_frontend_start: failed to start kernel_thread (%d)\n", ret);
+ fe_thread = kthread_run(dvb_frontend_thread, fe,
+  "kdvb-fe-%i", fe->dvb->num);
+ if (IS_ERR(fe_thread)) {
+  ret = PTR_ERR(fe_thread);
+  printk("dvb_frontend_start: failed to start kthread (%d)\n", ret);
   up(&fepriv->sem);
   return ret;
  }
- fepriv->thread_pid = ret;
-
+ fepriv->thread = fe_thread;
  return 0;
 }

Index: 2.6.19-mm1/drivers/media/dvb/ttpci/av7110.c
===================================================================
--- 2.6.19-mm1.orig/drivers/media/dvb/ttpci/av7110.c
+++ 2.6.19-mm1/drivers/media/dvb/ttpci/av7110.c
@@ -51,6 +51,7 @@
 #include <linux/firmware.h>
 #include <linux/crc32.h>
 #include <linux/i2c.h>
+#include <linux/kthread.h>

 #include <asm/system.h>

@@ -223,11 +224,10 @@ static void recover_arm(struct av7110 *a

 static void av7110_arm_sync(struct av7110 *av7110)
 {
- av7110->arm_rmmod = 1;
- wake_up_interruptible(&av7110->arm_wait);
+ if (av7110->arm_thread)
+  kthread_stop(av7110->arm_thread);

- while (av7110->arm_thread)
-  msleep(1);
+ av7110->arm_thread = NULL;
 }

 static int arm_thread(void *data)
@@ -238,17 +238,11 @@ static int arm_thread(void *data)
```

```
  dprintk(4, "%p\n",av7110);

- lock_kernel();
- daemonize("arm_mon");
- sigfillset(&current->blocked);
- unlock_kernel();
-
- av7110->arm_thread = current;
-
  for (;;) {
   timeout = wait_event_interruptible_timeout(av7110->arm_wait,
-          av7110->arm_rmmod, 5 * HZ);
-  if (-ERESTARTSYS == timeout || av7110->arm_rmmod) {
+   kthread_should_stop(), 5 * HZ);
+
+  if (-ERESTARTSYS == timeout || kthread_should_stop()) {
    /* got signal or told to quit*/
    break;
   }
@@ -276,7 +270,6 @@ static int arm_thread(void *data)
   av7110->arm_errors = 0;
  }

- av7110->arm_thread = NULL;
  return 0;
 }

@@ -2338,6 +2331,7 @@ static int __devinit av7110_attach(struc
  const int length = TS_WIDTH * TS_HEIGHT;
  struct pci_dev *pdev = dev->pci;
  struct av7110 *av7110;
+ struct task_struct *thread;
  int ret, count = 0;

  dprintk(4, "dev: %p\n", dev);
@@ -2622,9 +2616,12 @@ static int __devinit av7110_attach(struc
   printk ("dvb-ttpci: Warning, firmware version 0x%04x is too old. "
    "System might be unstable!\n", FW_VERSION(av7110->arm_app));

- ret = kernel_thread(arm_thread, (void *) av7110, 0);
- if (ret < 0)
+ thread = kthread_run(arm_thread, (void *) av7110, "arm_mon");
+ if (IS_ERR(thread)) {
+  ret = PTR_ERR(thread);
   goto err_stop_arm_9;
+ }
+ av7110->arm_thread = thread;
```

```
 /* set initial volume in mixer struct */
 av7110->mixer.volume_left  = volume;
Index: 2.6.19-mm1/drivers/media/dvb/ttpci/av7110.h
===================================================================
--- 2.6.19-mm1.orig/drivers/media/dvb/ttpci/av7110.h
+++ 2.6.19-mm1/drivers/media/dvb/ttpci/av7110.h
@@ -205,7 +205,6 @@ struct av7110 {
 struct task_struct *arm_thread;
 wait_queue_head_t   arm_wait;
 u16      arm_loops;
- int      arm_rmmod;

 void     *debi_virt;
 dma_addr_t     debi_bus;
```

____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers