
Subject: [RFC, PATCH] handle the multi-threaded init's exit() properly
Posted by [Oleg Nesterov](#) on Thu, 02 Aug 2007 21:20:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

With or without this patch, multi-threaded init's are not fully supported, but do_exit() is completely wrong. This becomes a real problem when we support pid namespaces.

1. do_exit() panics when the main thread of /sbin/init exits. It should not until the whole thread group exits. Move the code below, under the "if (group_dead)" check.

Note: this means that forget_original_parent() can use an already dead child_reaper()'s task_struct. This is OK for /sbin/init because

- do_wait() from alive sub-thread still can reap a zombie, we iterate over all sub-thread's ->children lists
- do_notify_parent() will wakeup some alive sub-thread because it sends the group-wide signal

However, we should remove choose_new_parent()->BUG_ON(reaper->exit_state) for this.

2. We are playing games with ->nsproxy->pid_ns. This code is bogus today, and it has to be changed anyway when we really support pid namespaces, just remove it.

Signed-off-by: Oleg Nesterov <oleg@tv-sign.ru>

```
--- t/kernel/exit.c~ 2007-08-03 00:10:28.000000000 +0400
+++ t/kernel/exit.c 2007-08-03 01:12:18.000000000 +0400
@@ -604,11 +604,6 @@ static void exit_mm(struct task_struct *
static inline void
choose_new_parent(struct task_struct *p, struct task_struct *reaper)
{
- /*
- * Make sure we're not reparenting to ourselves and that
- * the parent is not a zombie.
- */
- BUG_ON(p == reaper || reaper->exit_state);
  p->real_parent = reaper;
}

@@ -895,6 +890,14 @@ static void check_stack_usage(void)
static inline void check_stack_usage(void) {}
#endif
```

```

+static inline void exit_child_reaper(struct task_struct *tsk)
+{
+ if (likely(tsk->group_leader != child_reaper(tsk)))
+ return;
+
+ panic("Attempted to kill init!");
+}
+
fastcall NORET_TYPE void do_exit(long code)
{
    struct task_struct *tsk = current;
@@ -908,13 +911,6 @@ fastcall NORET_TYPE void do_exit(long co
    panic("Aiee, killing interrupt handler!");
    if (unlikely(!tsk->pid))
        panic("Attempted to kill the idle task!");
- if (unlikely(tsk == child_reaper(tsk))) {
- if (tsk->nsproxy->pid_ns != &init_pid_ns)
- tsk->nsproxy->pid_ns->child_reaper = init_pid_ns.child_reaper;
- else
- panic("Attempted to kill init!");
- }
-

    if (unlikely(current->ptrace & PT_TRACE_EXIT)) {
        current->ptrace_message = code;
@@ -964,6 +960,7 @@ fastcall NORET_TYPE void do_exit(long co
    }
    group_dead = atomic_dec_and_test(&tsk->signal->live);
    if (group_dead) {
+ exit_child_reaper(tsk);
    hrtimer_cancel(&tsk->signal->real_timer);
    exit_itimers(tsk->signal);
    }

```

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>

Subject: Re: [RFC, PATCH] handle the multi-threaded init's exit() properly

Posted by [akpm](#) on Mon, 06 Aug 2007 20:13:45 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Fri, 3 Aug 2007 01:20:09 +0400 Oleg Nesterov <oleg@tv-sign.ru> wrote:

> With or without this patch, multi-threaded init's are not fully supported, but
> do_exit() is completely wrong. This becomes a real problem when we support pid

```

> namespaces.
>
> 1. do_exit() panics when the main thread of /sbin/init exits. It should not
>    until the whole thread group exits. Move the code below, under the
>    "if (group_dead)" check.
>
>    Note: this means that forget_original_parent() can use an already dead
>    child_reaper()'s task_struct. This is OK for /sbin/init because
>
>    - do_wait() from alive sub-thread still can reap a zombie, we iterate
>      over all sub-thread's ->children lists
>
>    - do_notify_parent() will wakeup some alive sub-thread because it sends
>      the group-wide signal
>
>    However, we should remove choose_new_parent()->BUG_ON(reaper->exit_state)
>    for this.
>
> 2. We are playing games with ->nsproxy->pid_ns. This code is bogus today, and
>    it has to be changed anyway when we really support pid namespaces, just
>    remove it.
>
> Signed-off-by: Oleg Nesterov <oleg@tv-sign.ru>
>
> --- t/kernel/exit.c~ 2007-08-03 00:10:28.000000000 +0400
> +++ t/kernel/exit.c 2007-08-03 01:12:18.000000000 +0400
> @@ -604,11 +604,6 @@ static void exit_mm(struct task_struct *
> static inline void
> choose_new_parent(struct task_struct *p, struct task_struct *reaper)
> {
> - /*
> -  * Make sure we're not reparenting to ourselves and that
> -  * the parent is not a zombie.
> -  */
> - BUG_ON(p == reaper || reaper->exit_state);
>   p->real_parent = reaper;
> }
>
> @@ -895,6 +890,14 @@ static void check_stack_usage(void)
> static inline void check_stack_usage(void) {}
> #endif
>
> +static inline void exit_child_reaper(struct task_struct *tsk)
> +{
> + if (likely(tsk->group_leader != child_reaper(tsk)))
> + return;
> +
> + panic("Attempted to kill init!");

```

```

> +}
> +
> fastcall NORET_TYPE void do_exit(long code)
> {
>   struct task_struct *tsk = current;
> @@ -908,13 +911,6 @@ fastcall NORET_TYPE void do_exit(long co
>   panic("Aiee, killing interrupt handler!");
>   if (unlikely(!tsk->pid))
>     panic("Attempted to kill the idle task!");
> - if (unlikely(tsk == child_reaper(tsk))) {
> -   if (tsk->nsproxy->pid_ns != &init_pid_ns)
> -     tsk->nsproxy->pid_ns->child_reaper = init_pid_ns.child_reaper;
> - else
> -   panic("Attempted to kill init!");
> - }
> -
>
>   if (unlikely(current->ptrace & PT_TRACE_EXIT)) {
>     current->ptrace_message = code;
> @@ -964,6 +960,7 @@ fastcall NORET_TYPE void do_exit(long co
>   }
>   group_dead = atomic_dec_and_test(&tsk->signal->live);
>   if (group_dead) {
> +   exit_child_reaper(tsk);
>   hrtimer_cancel(&tsk->signal->real_timer);
>   exit_itimers(tsk->signal);
>   }

```

This patch broke

pid-namespaces-define-and-use-task_active_pid_ns-wrapper.patch. This hunk:

```

*****
*** 908,915 ****
    if (unlikely(!tsk->pid))
        panic("Attempted to kill the idle task!");
    if (unlikely(tsk == child_reaper(tsk))) {
-   if (tsk->nsproxy->pid_ns != &init_pid_ns)
-     tsk->nsproxy->pid_ns->child_reaper = init_pid_ns.child_reaper;
    else
        panic("Attempted to kill init!");
    }
--- 908,916 ----
    if (unlikely(!tsk->pid))
        panic("Attempted to kill the idle task!");
    if (unlikely(tsk == child_reaper(tsk))) {
+   if (task_active_pid_ns(tsk) != &init_pid_ns)
+     task_active_pid_ns(tsk)->child_reaper =

```

```
+   init_pid_ns.child_reaper;
    else
        panic("Attempted to kill init!");
}
```

has no place to live any more, so I just removed it.

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

Subject: Re: [RFC, PATCH] handle the multi-threaded init's exit() properly
Posted by [Oleg Nesterov](#) on Mon, 06 Aug 2007 20:33:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 08/06, Andrew Morton wrote:

```
>
> On Fri, 3 Aug 2007 01:20:09 +0400 Oleg Nesterov <oleg@tv-sign.ru> wrote:
>
> > 2. We are playing games with ->nsproxy->pid_ns. This code is bogus today, and
> > it has to be changed anyway when we really support pid namespaces, just
> > remove it.
>
> This patch broke
>
> pid-namespaces-define-and-use-task_active_pid_ns-wrapper.patch. This hunk:
>
> *****
> *** 908,915 ****
>  if (unlikely(!tsk->pid))
>  panic("Attempted to kill the idle task!");
>  if (unlikely(tsk == child_reaper(tsk))) {
> -  if (tsk->nsproxy->pid_ns != &init_pid_ns)
> -  tsk->nsproxy->pid_ns->child_reaper = init_pid_ns.child_reaper;
>  else
>  panic("Attempted to kill init!");
>  }
> --- 908,916 ----
>  if (unlikely(!tsk->pid))
>  panic("Attempted to kill the idle task!");
>  if (unlikely(tsk == child_reaper(tsk))) {
> +  if (task_active_pid_ns(tsk) != &init_pid_ns)
> +  task_active_pid_ns(tsk)->child_reaper =
> +  init_pid_ns.child_reaper;
>  else
>  panic("Attempted to kill init!");
```

> }
>
> has no place to live any more, so I just removed it.

Ah, thanks. I should have done this patch against -mm tree.

I hope it is OK to drop this chunk of
pid-namespaces-define-and-use-task_active_pid_ns-wrapper.patch

Because it can't work right now anyway, and Sukadev+Pavel already have
new patches on top this one which make namespace switch actually work.

Oleg.

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

Subject: Re: [RFC, PATCH] handle the multi-threaded init's exit() properly
Posted by [akpm](#) on Tue, 07 Aug 2007 05:18:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Tue, 7 Aug 2007 00:33:26 +0400 Oleg Nesterov <oleg@tv-sign.ru> wrote:

> On 08/06, Andrew Morton wrote:
> >
> > On Fri, 3 Aug 2007 01:20:09 +0400 Oleg Nesterov <oleg@tv-sign.ru> wrote:
> > >
> > > 2. We are playing games with ->nsproxy->pid_ns. This code is bogus today, and
> > > it has to be changed anyway when we really support pid namespaces, just
> > > remove it.
> >
> > This patch broke
> >
> > pid-namespaces-define-and-use-task_active_pid_ns-wrapper.patch. This hunk:
> >
> > *****
> > *** 908,915 ****
> > if (unlikely(!tsk->pid))
> > panic("Attempted to kill the idle task!");
> > if (unlikely(tsk == child_reaper(tsk))) {
> > - if (tsk->nsproxy->pid_ns != &init_pid_ns)
> > - tsk->nsproxy->pid_ns->child_reaper = init_pid_ns.child_reaper;
> > else
> > panic("Attempted to kill init!");
> > }

```

> > --- 908,916 ----
> >  if (unlikely(!tsk->pid))
> >  panic("Attempted to kill the idle task!");
> >  if (unlikely(tsk == child_reaper(tsk))) {
> > +  if (task_active_pid_ns(tsk) != &init_pid_ns)
> > +  task_active_pid_ns(tsk)->child_reaper =
> > +  init_pid_ns.child_reaper;
> >  else
> >  panic("Attempted to kill init!");
> >  }
> >
> > has no place to live any more, so I just removed it.
>
> Ah, thanks. I should have done this patch against -mm tree.
>
> I hope it is OK to drop this chunk of
> pid-namespaces-define-and-use-task_active_pid_ns-wrapper.patch
>
> Because it can't work right now anyway, and Sukadev+Pavel already have
> new patches on top this one which make namespace switch actually work.
>

```

OK, well I had to make a bit of on-the-fly adjustment to
pid-namespaces-rename-child_reaper-function.patch as well.

The diff-of-the-diff is:

```

@@ -48,7 +48,7 @@
diff -puN kernel/exit.c~pid-namespaces-rename-child_reaper-function kernel/exit.c
--- a/kernel/exit.c~pid-namespaces-rename-child_reaper-function
+++ a/kernel/exit.c
-@@ -694,7 +694,7 @@ forget_original_parent(struct task_struct
+@@ -683,7 +683,7 @@ forget_original_parent(struct task_struct
do {
    reaper = next_thread(reaper);
    if (reaper == father) {
@@ -57,19 +57,10 @@
    break;
}
} while (reaper->exit_state);
-@@ -907,7 +907,7 @@ fastcall NORET_TYPE void do_exit(long co
-  panic("Aiee, killing interrupt handler!");
-  if (unlikely(!tsk->pid))
-  panic("Attempted to kill the idle task!");
-- if (unlikely(tsk == child_reaper(tsk))) {
-+ if (unlikely(tsk == task_child_reaper(tsk))) {
-  if (task_active_pid_ns(tsk) != &init_pid_ns)
-  task_active_pid_ns(tsk)->child_reaper =

```

- init_pid_ns.child_reaper;
diff -puN kernel/signal.c~pid-namespaces-rename-child_reaper-function kernel/signal.c

Hopefully people can re-review and retest what's there in next -mm.

Or if that's too much work or too risky, option b) is to drop handle-the-multi-threaded-inits-exit-properly.patch, go back to the 2.6.23-rc1-mm2 versions of pid-namespaces-define-and-use-task_active_pid_ns-wrapper.patch and pid-namespaces-rename-child_reaper-function.patch and to ask Oleg to cook a 2.6.23-rc1-mm2 version of handle-the-multi-threaded-inits-exit-properly.patch.

The downside of this approach is that handle-the-multi-threaded-inits-exit-properly.patch looks more 2.6.24-ready than all the container stuff (based just on overall impact and speculativeness)

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

Subject: Re: [RFC, PATCH] handle the multi-threaded init's exit() properly
Posted by [Sukadev Bhattiprolu](#) on Tue, 07 Aug 2007 06:34:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Andrew Morton [akpm@linux-foundation.org] wrote:

| On Tue, 7 Aug 2007 00:33:26 +0400 Oleg Nesterov <oleg@tv-sign.ru> wrote:

|
| > On 08/06, Andrew Morton wrote:
| > >
| > > On Fri, 3 Aug 2007 01:20:09 +0400 Oleg Nesterov <oleg@tv-sign.ru> wrote:
| > >
| > > > 2. We are playing games with ->nsproxy->pid_ns. This code is bogus today, and
| > > > it has to be changed anyway when we really support pid namespaces, just
| > > > remove it.
| > >
| > > This patch broke
| > >
| > > pid-namespaces-define-and-use-task_active_pid_ns-wrapper.patch. This hunk:
| > >
| > > *****
| > > *** 908,915 ***
| > > if (unlikely(!tsk->pid))
| > > panic("Attempted to kill the idle task!");
| > > if (unlikely(tsk == child_reaper(tsk))) {
| > > - if (tsk->nsproxy->pid_ns != &init_pid_ns)

```

| >> - tsk->nsproxy->pid_ns->child_reaper = init_pid_ns.child_reaper;
| >> else
| >> panic("Attempted to kill init!");
| >> }
| >> --- 908,916 ----
| >> if (unlikely(!tsk->pid))
| >> panic("Attempted to kill the idle task!");
| >> if (unlikely(tsk == child_reaper(tsk))) {
| >> + if (task_active_pid_ns(tsk) != &init_pid_ns)
| >> + task_active_pid_ns(tsk)->child_reaper =
| >> + init_pid_ns.child_reaper;
| >> else
| >> panic("Attempted to kill init!");
| >> }
| >>
| >> has no place to live any more, so I just removed it.
| >
| > Ah, thanks. I should have done this patch against -mm tree.
| >
| > I hope it is OK to drop this chunk of
| > pid-namespaces-define-and-use-task_active_pid_ns-wrapper.patch
| >
| > Because it can't work right now anyway, and Sukadev+Pavel already have
| > new patches on top this one which make namespace switch actually work.
| >
|
| OK, well I had to make a bit of on-the-fly adjustment to
| pid-namespaces-rename-child_reaper-function.patch as well.
|
| The diff-of-the-diff is:
|
| @@ -48,7 +48,7 @@
| diff -puN kernel/exit.c~pid-namespaces-rename-child_reaper-function kernel/exit.c
| --- a/kernel/exit.c~pid-namespaces-rename-child_reaper-function
| +++ a/kernel/exit.c
| -@@ -694,7 +694,7 @@ forget_original_parent(struct task_struct
| +@@ -683,7 +683,7 @@ forget_original_parent(struct task_struct
| do {
|     reaper = next_thread(reaper);
|     if (reaper == father) {
| @@ -57,19 +57,10 @@
|     break;
| }
| } while (reaper->exit_state);
| -@@ -907,7 +907,7 @@ fastcall NORET_TYPE void do_exit(long co
| - panic("Aieee, killing interrupt handler!");
| - if (unlikely(!tsk->pid))
| - panic("Attempted to kill the idle task!");

```

```
| -- if (unlikely(tsk == child_reaper(tsk))) {  
| + if (unlikely(tsk == task_child_reaper(tsk))) {  
| - if (task_active_pid_ns(tsk) != &init_pid_ns)  
| - task_active_pid_ns(tsk)->child_reaper =  
| - init_pid_ns.child_reaper;  
| diff -puN kernel/signal.c~pid-namespaces-rename-child_reaper-function kernel/signal.c
```

| Hopefully people can re-review and retest what's there in next -mm.

| Or if that's too much work or too risky, option b) is to drop
| handle-the-multi-threaded-inits-exit-properly.patch, go back to the
| 2.6.23-rc1-mm2 versions of
| pid-namespaces-define-and-use-task_active_pid_ns-wrapper.patch and
| pid-namespaces-rename-child_reaper-function.patch and to ask Oleg to cook a 2.6.23-rc1-mm2
| version of handle-the-multi-threaded-inits-exit-properly.patch.

| The downside of this approach is that
| handle-the-multi-threaded-inits-exit-properly.patch looks more 2.6.24-ready
| than all the container stuff (based just on overall impact and
| speculativeness)

Well, if it will help, we can drop these two patches from -mm, take Oleg's
patch and I can then resend these along with other pid ns patches.

pid-namespaces-define-and-use-task_active_pid_ns-wrapper.patch.
pid-namespaces-rename-child_reaper-function.patch
Suka

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

Subject: Re: [RFC, PATCH] handle the multi-threaded init's exit() properly
Posted by [akpm](#) on Tue, 07 Aug 2007 07:32:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Mon, 6 Aug 2007 23:34:31 -0700 sukadev@us.ibm.com wrote:

> | The downside of this approach is that
> | handle-the-multi-threaded-inits-exit-properly.patch looks more 2.6.24-ready
> | than all the container stuff (based just on overall impact and
> | speculativeness)
>
> Well, if it will help, we can drop these two patches from -mm, take Oleg's
> patch and I can then resend these along with other pid ns patches.
>
> pid-namespaces-define-and-use-task_active_pid_ns-wrapper.patch.

> pid-namespaces-rename-child_reaper-function.patch

I'm reasonably sure that what I ended up with is OK. I'll send those two out again for double-checking, please.

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>
