Subject: Re: [PATCH 1/15] Move exit_task_namespaces()
Posted by Oleg Nesterov on Mon, 06 Aug 2007 13:55:53 GMT
View Forum Message <> Reply to Message

On 08/06, Pavel Emelyanov wrote:
>
> Oleg Nesterov wrote:
> >On 08/06, Pavel Emelyanov wrote:
> >>Oleg Nesterov wrote:
> >>>On 08/06, Pavel Emelyanov wrote:
> >>>>Oleg Nesterov wrote:
> >>>>>On 08/06, Pavel Emelyanov wrote:
> >>>>>>Oleg Nesterov wrote:
> >>>>>>>On 07/26, Pavel Emelyanov wrote:
> >>>>>>>>The reason to release namespaces after reparenting is that when task
> >>>>>>>>exits it may send a signal to its parent (SIGCHLD), but if the
> >>>>>>>>parent
> >>>>>>>>has already exited its namespaces there will be no way to decide
> >>>>>>>>what
> >>>>>>>>pid to dever to him - parent can be from different namespace.
> >>>>>>>I almost forgot about this one...
> >>>>>>>
> >>>I guess I missed something stupid and simple...
> >>In other words. Let task X live in init_pid_ns, task Y is his child and
> >>lives
> >>int another namespace. task X and task Y both die. This will happen:
> >>
> >>1. Task X call exit_task_namespaces()
> >> and sets its nsproxy to NULL
> >
> >Ah, got it, thanks. So the problem is not namespace itself (parent's or
> >child's), there are still valid (even if different but related).
> >
> >We just can't get ->parent->nsproxy. I was greatly confused by the "parent
> >can be from different namespace" above. We have exactly same problem if
> >namespaces are not differ.
> >
> >IOW, the problem is: we can't clear ->nsproxy (exit_task_namespaces) until
> >we get rid of ->children. This have nothing to do with different namespace.
>
> No. If the parent is always in the same namespace we do not need to
> get its nsproxy :) Problem is exactly in that the parent's namespace
> is to be known.

Yes yes, I see. I meant: once do_notify_parent() was modified to use
parent->nsproxy to figure out correct pid_t, that problem has nothing
to do with namespaces, it is just parent->nsproxy access.

But this is not safe, btw? do_notify_parent() can get parent->nsproxy
which is under destruction (sys_unshare). Then we read its ->pid_ns,
but at this time "struct nsproxy" could be kmem_cache_free()'ed ?

Of course, this is just theoretical, irqs are disabled, and the window
is tiny.

Oleg.
