
Subject: Re: [PATCH 2/2] pidns: Support unsharing the pid namespace.
Posted by [Oleg Nesterov](#) on Tue, 15 Feb 2011 19:01:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 02/15, Daniel Lezcano wrote:

>
> - Pass both nsproxy->pid_ns and task_active_pid_ns to copy_pid_ns
> As they can now be different.

But since they can be different we have to convert some users of
current->nsproxy first? But that patch was dropped.

> Unsharing of the pid namespace unlike unsharing of other namespaces
> does not take effect immediately. Instead it affects the children
> created with fork and clone.

IOW, unshare(CLONE_NEWPID) implicitly affects the subsequent fork(),
using the very subtle way.

I have to admit, I can't say I like this very much. OK, if we need
this, can't we just put something into, say, signal->flags so that
copy_process can check and create the new namespace.

Also. I remember, I already saw something like this and google found
my questions. I didn't actually read the new version, perhaps my
concerns were already answered...

But what if the task T does unshare(CLONE_NEWPID) and then, say,
pthread_create() ? Unless I missed something, the new thread won't
be able to see T ?

and, in this case the exiting sub-namespace init also kills its
parent?

OK, suppose it does fork() after unshare(), then another fork().
In this case the second child lives in the same namespace with
init created by the 1st fork, but it is not descendant ? This means
in particular that if the new init exits, zap_pid_ns_processes()->
do_wait() can't work.

Or not?

Oleg.

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