
Subject: Re: Pid namespace patchsets review

Posted by [Herbert Poetzl](#) on Sun, 11 Mar 2007 14:26:30 GMT

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

On Sat, Mar 10, 2007 at 06:57:13PM -0700, Eric W. Biederman wrote:

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

>

> > IMHO not the best idea, mainly because both OpenVZ

> > and Linux-VServer will end up either duplicating

> > the pid code or using the incomplete (broken) version

> > which probably gives the pid space a bad start ...

> >

> > I'd prefer to focus on fixing up the existing pid

> > issues (conversion) first, then hitting it with a

> > hopefully working pid namespace ...

> >

> > YMMV

>

> Right now if we discount the kernel_thread to kthread conversion

> we are probably 98% done with all of the conversions that make sense

> without a pid namespace.

>

> I guess NFS is the a big one still on the todo list.

>

> The point is that there are only a handful of things that we know

> about that we still need to convert that make a difference in

> practice.

>

> In addition the semantics of the pid namespace make a very big

> difference in understanding how we need to group processes. Having

> code people can look at and play with makes the subject a lot more

> approachable.

>

> Most of the remaining conversions do not actually make sense without

> the pid namespace so we have work to do there.

>

> Largely I am trying to structure this in a fashion that is accessible

> to more people, which means more people can work on it together.

>

>

> I think it would be reasonable to not merge the patch that enables

> clone/unshare support upstream until we have everything else finished.

>

> I have no intention of declaring a pid namespace done or complete

> until it is but getting as close as we can get would be a real

> advantage.

sure, I'm perfectly fine with keeping all that stuff

in -mm and test the hell out of it ... no problem
to make a new Linux-VServer branch based on -mm which
provides folks interested in testing to exercise the
pid namespace stuff ...

just in mainline, it would be a bad idea (IMHO)

> >> - When we do the rename can we please rename it task_proxy and
> >> have the functions follow that naming. The resource limiting
> >> conversation seems to be going in that direction, and it more
> >> general then what we are using now.
> >
> > hmm, nsproxy was unusual but kind of understandable,
> > task_proxy sounds just weird to me, I'd definitely
> > prefer nsproxy over task_proxy, but I'm open for
> > more 'space' related names too, like spaces or
> > space_proxy or space_group ...
>
> Well it is a proxy for task_struct and task_struct_proxy is just
> long winded. Calling it task_proxy makes sticking the pointers
> to other subsystems per task data more reasonable.

interesting view, for me it always was a proxy for
the (name)spaces, as for me, the direction always
is task -> proxy -> space, not the other way round

but I'm not going to insist in naming that differently
(i.e. if the majority finds that naming intuitive,
I'm fine getting myself used to it)

best,
Herbert

> Eric

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