Subject: Re: [RFC][PATCH] rename 'struct pid'
Posted by Dave Hansen on Wed, 11 Apr 2007 21:28:47 GMT
View Forum Message <> Reply to Message

On Wed, 2007-04-11 at 14:54 -0600, Eric W. Biederman wrote:
> Dave Hansen <hansendc@us.ibm.com> writes:
>
> > On Wed, 2007-04-11 at 12:46 -0600, Eric W. Biederman wrote:
> >>
> >> > These can be a bit confusing:
> >> >
> >> >       struct pid *pid;
> >> >       struct pid *pgrp;
> >> >       struct pid *sid;
> >>
> >> How is it more confusing then?
> >>
> >>         pid_t pid;
> >>         pid_t pgrp;
> >>         pid_t sid;
> >
> > They confuse me the same way. :)
> >
> > We can't do much about userspace.  But, we do have quite a bit of
> > control how we name things in the kernel, and I think there's a better
> > way.
>
> Maybe.
>
> The worst of those above is:
> pid_t pid;
>
> Am I correct?

Definitely.

> When someone mentions a pid which side of the above statement are you
> thinking of the left hand side or the right hand side.  The type or
> the variable name.

Traditionally, I think of a pid as what I see in top.  So, I think of
the right hand side variable name.  I think of it this way because the
left hand side has little meaning in how the pid_t is going to be used.

>  If the issue is that you find the concept of pid_t confusing then it
>  is much harder to sort this out.

I find pid_t confusing.  There, I've said it. ;)

In a perfect world, kill() wouldn't be multiplexed the way it is. We'd have kill_myself(), kill_pgrp(pgrp), kill_pid() and the pid_t passed into kill_pid() there would only mean 'process id', only and could _never_ mean 'process group id'.

We could even have different data structures so that type safety would keep get_pgrp()'s result from being easily fed to kill_pgrp().

-- Dave

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