Subject: Re: [RFC][PATCH] rename 'struct pid'
Posted by ebiederm on Wed, 11 Apr 2007 01:28:02 GMT
View Forum Message <> Reply to Message

Dave Hansen <hansendc@us.ibm.com> writes:

> I've been hacking quite a bit on the pidspace code. I've run
> into a bug or two, and had a heck of a time debugging it.
> Other than my inferior puny monkey brain, I'm directing some
> of the blame squarely in the direction of the 'struct pid'.
>
> We have pid_t, pid_ns, struct pid and pid_link, at _least_.
> Seeing code like get_pid(pid->pid_ns->pid_type[PIDTYPE_PID])
> is mind-numbing to say the least.

get_pid(pid->pid_ns->pid_type[PIDTYPE_PID]) is complete and utter
nonsense.

> It makes it really hard to comprehend, and even harder to debug.

Given that you quoted nonsense I can understand the comprehension
problem.

> We honestly have a lot of code like this:
>
> pid = pid_nr(filp->f_owner.pid);
>
> WTF? It's getting a pid from a pid? Huh? :)

Clearer would be:

user_pid = pid_to_user(filp->f_owner.pid);

> It makes sense when you go look at the structures, but
> sitting in the middle of a function and when you can't see
> all of the struct declarations can be really sketchy.
>
> So, I propose that we rename the one structure that seems to
> be the focus of the problem: 'struct pid'.

NAK.

> Fundamentally, it
> is a 'process identifier': it helps the kernel to identifty
> processes. However, as I noted, 'pid' is a wee bit overloaded.
>
> In addition to "identifying" processes, this structure acts

> as an indirection or handle to them. So, I propse we call
> these things 'struct task_ref'.

Renaming the structure above doesn't help and the structure represents
a pid in a more fundamental way then pid_t can.   A pid (pid_t or
struct pid) isn't just an identifier it is a handle to processes.
struct pid just does so more directly because it is inside the kernel.

> Just reading some of the
> code that I've modified in here makes me feel like this is
> the right way.

I get exactly the opposite impression.

> Compare the two sentences below:
>
> Oh, I have a task_ref? What kind is it? Oh, it's a pgid
> reference because I have REFTYPE_PGID.
>
> Oh, I have a pid? What kind is it? Oh, it's a pid because
> I have PIDTYPE_PID.
>
> Which makes more sense?

Neither the questions are nonsense. The only reasonable question
is which kind of process am I using the pid to look for.

> So, this still needs some work converting some of the
> function names, but it compiles as-is. Any ideas for better
> names?

struct pid is properly named. It isn't even as fuzzy as struct
signal_struct.

All I can suggest is making a clearer distinction between user and
kernel pids. So maybe it could become struct kpid. Even there
I'm not certain it makes sense except in variable names.

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