Subject: Re: Re: [RFC][PATCH] rename 'struct pid'
Posted by xemul on Wed, 11 Apr 2007 06:55:00 GMT
View Forum Message <> Reply to Message

Dave Hansen wrote:
> On Tue, 2007-04-10 at 22:52 -0600, Eric W. Biederman wrote:
>> Dave Hansen <hansendc@us.ibm.com> writes:
>>>> A pid (pid_t or
>>>> struct pid) isn't just an identfier it is a handle to processes.
>>>> struct pid just does so more directly because it is inside the kernel.
>>> Let's face it, "pid" has a meaning. It's a number. It's what you
>>> kill(1). The meaning has been there for a long, long time. 'struct
>>> pid' is a completely different concept, and it's certainly more than
>>> "just a number".
>> Yes. "pid" has a meaning. The meaning is old and well established.
>> That meaning is not just a number, just like a file descriptor is not
>> just a number.
>
> That's a great example. Userspace fds are to 'struct file' as pids are
> to 'struct pid', right?
>
> I actually think 'struct file' is a pretty good name. Think of what
> do_sys_open() might look like if we called 'struct file' 'struct fd'
> instead and 'fdp' instead of 'filp'.
>
> We end up with lines like:
>
> fd_install(fd, fdp);
>
> Which makes it confusing which fd we're dealing with or what the 'fd_'
> in the name refers to, the 'fd' or the 'fdp'. It makes quite a bit of
> sense to have 'fd' and 'struct file' named quite distinctly.

Agree. int fd is a *file* descriptor, i.e. a number that describes
a file, which is a struct file essentially. That's the way pids must
be represented. E.g. the pid_t is a number, that references some
kernel-space object. This object is to be called somehow more
descriptive than just struct pid.

Maybe it's worth renaming struct pid into struct pid_struct to
represent the fact, that this is a pid, but also a structure?

>>> So, please consider that there are actual kernel developers hacking on
>>> this stuff who are having problem with it. The function of 'struct pid'
>>> is great, it's a wonderful concept. It just needs a slightly different
>>> name in order to more easily relate that concept to those that are
>>> trying to use it.
>>>

>>> If anyone can think of some more incremental ways to do this, or has
>>> other suggestions on how to make it more clear, I'm all ears.
>> So what I have seen are examples down in the guts of the pid hash
>> table that are problematic. And a few complaints about pid_nr.
>>
>> However the conversions I have done and I have looked at have just
>> been a drop in replacement for pid_t except for reference counting
>> issues. That to me at least is rather convincing.
>
> I think this is more indicative of the great design of 'struct pid' in
> concept. It truly is a drop-in replacement for how things were used in
> the past. The concept is *great*.
>
>> So I'm not convinced there is a fundamental problem. Just a bit of a
>> problem in the guts of things where everything seems to have the
>> same name. I'm not at all certain how a different prefix would
>> sort that out.
>
> I agree that there's no fundamental problem with the structure. Its
> function is quite ideal. The issue for me comes in the ability for
> people to update, enhance and review what is going on. There's no
> fundamental barrier here, as Suka demonstrated getting some of his
> pidspace code to work. It just crossed my pain threshold as I was
> trying to debug some of it.
>
> Once we get pidspaces fully working, the hacking in the guts will
> certainly be reduced. But, there are always bugs, and this is a common
> enough code area that people are bound to touch it as time goes on. I
> just want to make that as easy as possible.
>
>> My feeling is that changing this just caters to people who are not
>> going to be able to understand what is going on no matter what the
>> structure is named, and is going to make it harder for me to update
>> the code when I find the time to do it.
>
> I'm completely sure that you'll grasp the entire concept, no matter to
> what we change the names. The mess that you've unraveled so far in
> there makes has given me supreme confidence in this. :)
>
> My worry is the ramp-up time for people who want to understand it enough
> hack it or just audit the code, but who won't grasp it on quite the same
> level that you have.
>
> -- Dave
>
> _____
> Containers mailing list
> Containers@lists.linux-foundation.org

Subject: Re: Re: [RFC][PATCH] rename 'struct pid'
Posted by dev on Wed, 11 Apr 2007 07:40:48 GMT

View Forum Message <> Reply to Message

Pavel Emelianov wrote:
> Dave Hansen wrote:
>
>>On Tue, 2007-04-10 at 22:52 -0600, Eric W. Biederman wrote:
>>
>>>Dave Hansen <hansendc@us.ibm.com> writes:
>>>
>>>>>A pid (pid_t or
>>>>>struct pid) isn't just an identfier it is a handle to processes.
>>>>>struct pid just does so more directly because it is inside the kernel.
>>>>
>>>>Let's face it, "pid" has a meaning. It's a number. It's what you
>>>>kill(1). The meaning has been there for a long, long time. 'struct
>>>>pid' is a completely different concept, and it's certainly more than
>>>>"just a number".
>>>
>>>Yes. "pid" has a meaning. The meaning is old and well established.
>>>That meaning is not just a number, just like a file descriptor is not
>>>just a number.
>>
>>That's a great example. Userspace fds are to 'struct file' as pids are
>>to 'struct pid', right?
>>
>>I actually think 'struct file' is a pretty good name. Think of what
>>do_sys_open() might look like if we called 'struct file' 'struct fd'
>>instead and 'fdp' instead of 'filp'.
>>
>>We end up with lines like:
>>
>> fd_install(fd, fdp);
>>
>>Which makes it confusing which fd we're dealing with or what the 'fd_'
>>in the name refers to, the 'fd' or the 'fdp'. It makes quite a bit of
>>sense to have 'fd' and 'struct file' named quite distinctly.
>>
>>
> Agree. int fd is a *file* descriptor, i.e. a number that describes
> a file, which is a struct file essentially. That's the way pids must
> be represented. E.g. the pid_t is a number, that references some
> kernel-space object. This object is to be called somehow more

> descriptive than just struct pid.
>
> Maybe it's worth renaming struct pid into struct pid_struct to
> represent the fact, that this is a pid, but also a structure?
it helps struct name only.
while fields should be renamed as well somehow.

Kirill