
Subject: Re: [RFC][PATCH] rename 'struct pid'
Posted by [ebiederm](#) on Thu, 12 Apr 2007 16:25:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

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

> On Wed, 2007-04-11 at 14:54 -0600, Eric W. Biederman wrote:

>> 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.

I think if you looked a little more closely you would be surprised.

>> 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'.

The multiplexing is an anomaly. kill(2) really is 4 different functions.

```
kill_process(pid_t process, int sig);  
kill_process_group(pid_t group, int sig);  
kill_current_process_group(int sig);  
kill_all(int sig);
```

And it is really an abuse of the data type to overload it quite they way it happens in kill(2).

However that no way in validates the fact that the identifiers for session, process groups, and processes are and should be of the same time.

To my knowledge the only kernel interface the brain damage of kill(2) propagates in a persistent way to is: "fcntl(F_GETOWN);" which is a tacked on interface. I admit there are a few others where the set of processes is interpreted the same as in kill(2) but on

fcntl actually reads back that way.

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

I assume here you mean to switch the type of process grouping, process
vs process group or something like that.

What tends to be instructive in the user space interface are the
definition of setsid() and setpgrp();

setsid() really is defined as: session = pgrp = getpid();
setpgrp() is just: pgrp = getpid();

It really is fundamental to unix use of pids that you can compare
and interchange the identifiers for sessions, process groups, and
processes.

Eric

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