Subject: Re: [RFC][PATCH] rename 'struct pid'
Posted by ebiederm on Wed, 11 Apr 2007 18:46:06 GMT
View Forum Message <> Reply to Message

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

> On Wed, 2007-04-11 at 11:34 -0600, Eric W. Biederman wrote:
>> A struct pid is very similar to a list symbol. Two of them can be compared
> for
>> equality just by comparing pointers. You can get different information about
>> a struct pid by examining different "slots". A struct pid has a
>> name, but unlike lisp the name is numeric instead of a string.
>
> I'm not sure what you mean by "list symbol". Is that a lisp concept?

Yes. It was the most similar concept I could think of.

> Userspace refers to processes, sessions, and process groups by numbers
> which are finite and can wrap over time. Instead of using these numbers
> inside the kernel, we use a 'struct pid'. Unlike a plain number, a
> 'struct pid' uniquely refers to a particular process, session, or
> process group, and does not suffer from any wrapping effects.

Just like the plain number a 'struct pid' simultaneously refers to a process,
session and a process group. Unlike a plain number it does not suffer any
wrapping effects.

> In general, the underlying task to which a 'struct pid' refers will not
> change. It is possible such as when a non-thread-leader does an exec()
> and takes over as the leader, tat the tasks to which a 'struct pid'
> refers will change.

There is no unique underlying task to which a 'struct pid' refers.

> This effectively lets the kernel do a pid_t (process, session, or
> process group) to task lookup at a particular time and keep the results
> of that lookup meaningful for a long, long time, not worrying about if
> userspace has re-used those values.
>
> ---
>
> I think at least part of the problem is that a 'struct pid' creates a
> relationship to pid, as well as session ids and process groups. This is
> a bit muddled relationship that comes out of userspace, but it would be
> nice to unmuddle it somehow in the kernel. I think part of that might
> be to give 'struct pid' a more neutral name that also encompasses the
> pgrp and sid parts.

Have you ever looked at which data type the number for a session and a process group are stored in?

> 'pref' seems a decent compromise. It keeps the "process" notion around
> (and thus connected to the userspace concepts) while helping to reduce
> the use of 'pid' in the name.
>
> 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;
```

> this is a bit better:
>
> struct pref *pid;
> struct pref *pgrp;
> struct pref *sid;

The reason I call it a pid is because it is. It is a pid in every way except the bloody numeric name.

A pid is an identifier for processes and groups of processes. Just which processes depends on the function you feed it into.

Eric