
Subject: Re: [RFC][PATCH 3/6] pid namespace : use struct pid_nr
Posted by [ebiederm](#) on Tue, 13 Mar 2007 08:52:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Cedric Le Goater <clg@fr.ibm.com> writes:

```
>>> Index: lx26-20-mm2b/kernel/pid.c
>>> =====
>>> --- lx26-20-mm2b.orig/kernel/pid.c 2007-03-09 15:29:21.000000000 -0800
>>> +++ lx26-20-mm2b/kernel/pid.c 2007-03-09 15:29:23.000000000 -0800
>>> @@ -180,8 +180,19 @@ fastcall void put_pid(struct pid *pid)
>>> if (!pid)
>>> return;
>>> if ((atomic_read(&pid->count) == 1) ||
>>> -    atomic_dec_and_test(&pid->count))
>>> +    atomic_dec_and_test(&pid->count)) {
>>> + struct pid_nr* pid_nr;
>>> + struct hlist_node *pos, *next;
>>> +
>>> + /*
>>> +  * rcu is not needed anymore
>>> +  */
>>>
>> rcu should never be needed...
>> We should be able to get away with a definition that is immutable for the
>> lifetime of a struct pid.
>
> but struct pid requires to be rcu safe and as the new struct pid_nr is a
> member of struct pid, it seems that the same rule should apply. nop ?
```

I was implying a strong property. A list that is created before we insert into the hash table and is destroyed after the last reference from the hash table is going.

So in that case we would have a property stronger than rcu.

The restrict would be that the list would be completely immutable, for the lifetime of the struct pid.

We may need to modify the list and do the whole rcu thing but I would prefer we not go in with that assumption until we know we need to. Modifying the mapping of struct pid into various pid namespaces adds a complexity to the user space interface that I would like to avoid.

The only practical use I can currently think of for modifying the struct pid list is when a namespace exists to cleanup our memory of what was in that namespace because pid_nr will no longer care.

Eric

Containers mailing list

Containers@lists.osdl.org

<https://lists.osdl.org/mailman/listinfo/containers>
