

---

Subject: Re: [PATCH 2/3] Pid ns helpers for signals  
Posted by [Sukadev Bhattiprolu](#) on Mon, 03 Sep 2007 16:55:14 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Oleg Nesterov [oleg@tv-sign.ru] wrote:

| On 08/31, sukadev@us.ibm.com wrote:

| >  
| > Define some helper functions that will be used to implement signal semantics  
| > with multiple pid namespaces.

| >  
| > is\_current\_in\_ancestor\_pid\_ns(task)  
| >  
| > TRUE iff active pid namespace of 'current' is an ancestor of  
| > active pid namespace of @task.

| >  
| > is\_current\_in\_same\_or\_ancestor\_pid\_ns(task)  
| >  
| > TRUE iff active pid namespace of 'current' is either same as  
| > or an ancestor of active pid namespace of @task.

| These names are awfull :) Yes, yes, it was me who suggested them... No, I can't  
| suggest something better.

I agree :) I tried smaller names like task\_ancestor\_pid\_ns() and passing in  
'current' as a parameter so its not in the name :) but the functionality was  
not obvious from the names.

|  
| > + \* Caller must hold a reference to @pid.  
| > + \*/  
| > +static inline struct pid\_namespace \*pid\_active\_ns(struct pid \*pid)  
| > +{  
| > + if (!pid)  
| > + return NULL;  
| > +  
| > + return pid->numbers[pid->level].ns;  
| > +}

| Well, the comment is a bit misleading. Yes, my previous comment was not very  
| clear. Yes, the function itself is not safe unless you know what are you doing,  
| like, for example, get\_pid(). I think it is better to just kill the comment.  
| Please see below.

Ok. will remove the comment.

|  
| > +static struct pid\_namespace \*get\_task\_pid\_ns(struct task\_struct \*tsk)  
| > +{  
| > + struct pid \*pid;

```

| > + struct pid_namespace *ns;
| > +
| > + pid = get_task_pid(tsk, PIDTYPE_PID);
| > + ns = get_pid_ns(pid_active_ns(pid));
| > + put_pid(pid);
| > +
| > + return ns;
| > +}

```

| Hmm. Firstly, we don't need this for the "current", but all users of this func  
| also do get\_task\_pid\_ns(current).

| Also, we don't need get/put\_pid. rcu locks are enough,

```

| rcu_read_lock();
| ns = get_pid_ns(pid_active_ns(task_pid(tsk)));
| rcu_read_unlock();
|

```

Ok.

| However, do we really need this complications right now? Currently, we use  
| this "compare namespaces" helpers only when we know that "struct pid" is  
| stable. We are sending the signal to that task, it must be pid\_alive(), and  
| we either locked the task itself, or we hold tasklist.

My concern was that the task could detach and free its pid which in turn  
would drop the last reference on a pid namespace and free it.

By trying to keep is\_current\_in\_ancestor\*() general, I guess it is more  
complicated than it needs to be right now.

Would holding the rcu\_read\_lock() be enough or since our callers hold  
it now, can we just drop that too ?

```
is_current_in_ancstor_pid_ns(tsk)
```

```

rcu_read_lock();
my_ns = pid_active_ns(current);
tsk_ns = pid_active_ns(tsk)
rc = is_ancestor_ns(my_ns, tsk_ns)
rcu_read_unlock();

```

```
return rc;
```

Thanks for the comments,

Suka

---

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

---