
Subject: Re: [PATCH 3/7] Containers (V8): Add generic multi-subsystem API to containers

Posted by [Srivatsa Vaddagiri](#) on Wed, 11 Apr 2007 05:01:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

[Sorry abt piece meal reviews, I am sending comments as and when I spot something]

On Fri, Apr 06, 2007 at 04:32:24PM -0700, menage@google.com wrote:

```
> -void container_exit(struct task_struct *tsk)
> +void container_exit(struct task_struct *tsk, int run_callbacks)
> {
```

[snip]

```
> + /* Reassign the task to the init_container_group. */
>   task_lock(tsk);
```

[snip]

```
> + if (tsk->containers != &init_container_group) {
```

Is this check needed? If we have the check, then:

```
> + task_unlock(tsk);
> + if (cg)
> +   put_container_group(cg);
```

init_container_group refcount is leaky (fork increments it, but exit doesn't). Not a big prob perhaps, but ..would be nice to avoid?

--

Regards,
vatsa

Subject: Re: [PATCH 3/7] Containers (V8): Add generic multi-subsystem API to containers

Posted by [Paul Menage](#) on Wed, 11 Apr 2007 08:42:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

On 4/10/07, Srivatsa Vaddagiri <vatsa@in.ibm.com> wrote:

```
> [ Sorry abt piece meal reviews, I am sending comments as and when I spot
> something ]
```

That's no problem.

>

> On Fri, Apr 06, 2007 at 04:32:24PM -0700, menage@google.com wrote:

```
> > -void container_exit(struct task_struct *tsk)
> > +void container_exit(struct task_struct *tsk, int run_callbacks)
> > {
>
> [snip]
>
> > + /* Reassign the task to the init_container_group. */
> >     task_lock(tsk);
>
> [snip]
>
> > + if (tsk->containers != &init_container_group) {
>
> Is this check needed? If we have the check, then:
>
> > +     task_unlock(tsk);
> > +     if (cg)
> > +         put_container_group(cg);
>
> init_container_group refcount is leaky (fork increments it, but exit
> doesnt). Not a big prob perhaps, but ..would be nice to avoid?
>
```

Yes, I think you're right - over a long uptime that would just cause problems ...

Paul
