Subject: Re: [RFC][PATCH] Make access to taks's nsproxy liter
Posted by Pavel Emelianov on Thu, 09 Aug 2007 07:12:19 GMT
View Forum Message <> Reply to Message

[snip]

>> diff --git a/include/linux/nsproxy.h b/include/linux/nsproxy.h
>> index 525d8fc..74f21fe 100644
>> --- a/include/linux/nsproxy.h
>> +++ b/include/linux/nsproxy.h
>> @@ -32,8 +32,14 @@ struct nsproxy {
>> };
>> extern struct nsproxy init_nsproxy;
>>
>> +static inline struct nsproxy *task_nsproxy(struct task_struct *tsk)
>> +{
>> + return rcu_dereference(tsk->nsproxy);
>> +}
>
> Looks like a very nice cleanup as well. But please add a comment
> above task_nsproxy() that it must be called under rcu_read_lock()
> or task_lock(task) (though I'll admit the rcu_dereference may make that
> obvious)

I will, but I think that rcu_dereference implies this. Anyway.

[snip]

>> + if (ns == new)
>> + return;
>> +
>> + if (new)
>> + get_nsproxy(new);
>> + rcu_assign_pointer(p->nsproxy, new);
>> +
>> + if (ns && atomic_dec_and_test(&ns->count)) {
>> + /*
>> + * wait for others to get what they want from this
>> + * nsproxy. cannot release this nsproxy via the
>> + * call_rcu() since put_mnt_ns will want to sleep
>> + */
>> + synchronize_rcu();
>> + free_nsproxy(ns);
>> + }
>> +}
>
> Also a comment above switch_task_namespaces() that it must be called
> with task_lock held.

no! no locks here! free_nsproxy() may sleep when putting mnt_ns and maybe some other. see - there's a hunk in sys_unshare that move the task_lock() after switch_task_namespaces().

> thanks,
> -serge

_____

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