Subject: Re: [PATCH][NETNS] Use list_for_each_entry_continue_reverse in setup_net
Posted by ebiederm on Fri, 14 Sep 2007 14:41:07 GMT

View Forum Message <> Reply to Message

Stephen Hemminger <shemminger@linux-foundation.org> writes:

> On Fri, 14 Sep 2007 11:39:32 +0400
> Pavel Emelyanov <xemul@openvz.org> wrote:
>
>> I proposed introducing a list_for_each_entry_continue_reverse
>> macro to be used in setup_net() when unrolling the failed
>> ->init callback.
>>
>> Here is the macro and some more cleanup in the setup_net() itself
>> to remove one variable from the stack :) Minor, but the code
>> looks nicer.
>>
>> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
>
> Maybe it is time to just eliminate the init hook from the API.
> It has very few users, and there is no reason the setup needed
> could be done before or after registering in most cases.

I guess only have 5 out of the 29 users I have in my full patchset
is few. But that is to be expected because so far only the core
has been converted.

I looked again at the initialization to see if you had a point about
the initialization but in every instance I looked at the function
was performing work that needed to happen during the creation of
each network namespace. So the work very much needs to be done there.

Ok looking some more I can see why this isn't obvious yet. copy_net_ns
hasn't been merged yet, and that is where we create new network namespaces.
And call setup_net on each new network namespace.

I will take a look at that patch and see if I can come up with a
safe version of it to merge to allow for a little more transparency.

```
> struct net *copy_net_ns(unsigned long flags, struct net *old_net)
> {
> 	struct net *new_net = NULL;
> 	int err;
>
> 	get_net(old_net);
>
> 	if (!(flags & CLONE_NEWNET))
>                 return old_net;
>
>         err = -EPERM;
>         if (!capable(CAP_SYS_ADMIN))
>                 goto out;
>
>         err = -ENOMEM;
>         new_net = net_alloc();
>         if (!new_net)
>                 goto out;
>
>         mutex_lock(&net_mutex);
>         err = setup_net(new_net);
>         if (err)
>                 goto out_unlock;
>
>         net_lock();
>         list_add_tail(&new_net->list, &net_namespace_list);
>         net_unlock();
>
>
> out_unlock:
>         mutex_unlock(&net_mutex);
> out:
>         put_net(old_net);
>         if (err) {
>                 net_free(new_net);
>                 new_net = ERR_PTR(err);
>         }
>         return new_net;
> }
```

Eric