

---

Subject: Re: [PATCH 03/16] net: Basic network namespace infrastructure.  
Posted by [ebiederm](#) on Mon, 10 Sep 2007 15:53:55 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Pavel Emelyanov <xemul@openvz.org> writes:

```
> Eric W. Biederman wrote:
>
> [snip]
>
>> --- /dev/null
>> +++ b/include/net/net_namespace.h
>> @@ -0,0 +1,68 @@
>> +/*
>> + * Operations on the network namespace
>> + */
>> +#ifndef __NET_NET_NAMESPACE_H
>> +#define __NET_NET_NAMESPACE_H
>> +
>> +#include <asm/atomic.h>
>> +#include <linux/workqueue.h>
>> +#include <linux/list.h>
>> +
>> +struct net {
>
> Isn't this name is too generic? Why not net_namespace?
```

My fingers went on strike. struct netns probably wouldn't be bad. The very long and spelled out version seemed painful. I don't really care except that not changing it is easier for me. I'm happy to do whatever is considered most maintainable.

```
>> + /* Walk through the list backwards calling the exit functions
>> + * for the pernet modules whose init functions did not fail.
>> + */
>> + for (ptr = ptr->prev; ptr != &pernet_list; ptr = ptr->prev) {
>
> Good reason for adding list_for_each_continue_reverse :)
```

Sounds reasonable.

```
>> +static int register_pernet_operations(struct list_head *list,
>> +      struct pernet_operations *ops)
>> +{
>> + struct net *net, *undo_net;
>> + int error;
>> +
>> + error = 0;
```

```
>> + list_add_tail(&ops->list, list);
>> + for_each_net(net) {
>> +   if (ops->init) {
>
> Maybe it's better to do it vice-versa?
> if (ops->init)
>   for_each_net(net)
>     ops->init(net);
> ...
```

My gut feel says it is more readable with the test on the inside.

Although something like

```
if (ops->init)
    goto out;
```

might be more readable.

```
>> +int register_pernet_device(struct pernet_operations *ops)
>> +{
>> + int error;
>> + mutex_lock(&net_mutex);
>> + error = register_pernet_operations(&pernet_list, ops);
>> + if (!error && (first_device == &pernet_list))
>
> Very minor: why do you give the name "device" to some pernet_operations?
```

Subsystems need to be initialized before and cleaned up after network devices. We don't have much in the way that needs this distinction, but we have just enough that it is useful to make this distinction.

Looking at my complete patchset all I have in this category is the loopback device, and it is important on the teardown side of things that the loopback device be unregistered before I clean up the protocols or else I get weird leaks.

Reflecting on it I'm not quite certain about the setup side of things. I'm on the fence if I need to completely dynamically allocate the loopback device or if I need to embed it struct net.

There also may be a call for some other special network devices to have one off instances in each network namespace. With the new netlink creation API it isn't certain we need that idiom anymore but before that point I was certain we would have other network devices besides the loopback that would care.

Eric

---

Containers mailing list

