Subject: Re: [PATCH 1/1] Dynamically allocate the loopback device
Posted by ebiederm on Mon, 27 Aug 2007 12:12:14 GMT
View Forum Message <> Reply to Message

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

> On Fri, 24 Aug 2007 19:55:47 +0400
> "Denis V. Lunev" <dlunev@gmail.com> wrote:
>
>> dlezcano@fr.ibm.com wrote:
>> > From: Daniel Lezcano <dlezcano@fr.ibm.com>
>> >
>> > Doing this makes loopback.c a better example of how to do a
>> > simple network device, and it removes the special case
>> > single static allocation of a struct net_device, hopefully
>> > making maintenance easier.
>> >
>> > Applies against net-2.6.24
>> >
>> > Tested on i386, x86_64
>> > Compiled on ia64, sparc
>>
>> I think that a small note, that initialization order is changed will be
>> good to record. After this, loopback MUST be allocated before any other
>> networking subsystem initialization. And this is an important change.
>>
>> Regards,
>> Den
>
> Yes, this code would break when other drivers are directly linked
> in.

No. Other drivers don't care at all about the loopback device,
and it isn't a requirement that the loopback device be initialized
before other devices.

The requirement is that the loopback device is allocated before
we start using it. Which means networking subsystems like
ipv4 and ipv6 care not other network drivers. In practices this means
before we get very far into the ipv4 subsystem initialization as
ipv4 is always compiled in and is initialized early.

To get the initialization order correct I used fs_initcall instead of
module_init.

When I reflect on it. I'm not really comfortable with the fact
that we currently start using the loopback device before we
finish initializing and register it. Although it has worked

for over a decade so I guess early on we don't care about
much more then the address of the loopback device.

>From what I can tell the initialization order dependency seems much
less subtle and much more robust then separate rules for allocating
the loopback device. We have had several patchs recently that
broke (including one merged upstream). The only way I can see
to break an initialization order dependency is to go deliberately
messing around with initialization order.

Eric

p.s. My apologies for the late reply some one dropped me off the cc.
And I have been under the weather all week.

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