

Re: adding if_dev member to struct ifnet

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2003-09/0149.html>

From: Poul-Henning Kamp (*phk_at_phk.freebsd.dk*)

Date: 09/30/03

To: John Baldwin <jhb@FreeBSD.org>
Date: Tue, 30 Sep 2003 17:08:34 +0200

In message <XFMail.20030930103348.jhb@FreeBSD.org>, John Baldwin writes:

>>>Yes, if it helps to remove if_name/if_unit, it is a thing to do. Moreover it
>>>sounds a good idea to have the if_dev field into the ifnet structure.
>>
>> Somebody please explain how this would work for non-hardware
>> interfaces like if_loop, if_tun, if_tap etc ?
>>
>> device_t is what we use to hitch drivers to hardware.
>>
>> ifnet is what we use to hitch drivers to the netstack.
>>
>> They should not be tangled.
>
>You mean like dev_t and device_t shouldn't be tangled like we do
>with si_drv1? Oh, wait...

I don't think any correctly written driver stores it's device_t in a dev_t. It should store it's softc structure, which should contain pointers to both. Even if there is a driver which does do that, it happens inside the device driver, and it does not handicap the remaining device drivers with its choice.

If you stick a newbus requirement on "struct ifnet" you suddenly make demands that a lot of our network drivers cannot satisfy.

The problem with propagating newbus above the device drivers is that we start postulating a specific relationship between logical devices and physical (ie: a ifnet has exactly one device_t).

There is nothing in the "data-model" of the kernel that says that a network interface corresponds to exactly one hardware device and more importantly: there shouldn't be either.

If_tun _has_ no physical device, and it would be totally insane to invent a device_t for if_tun, considering that it would not serve any purpose at all, apart from satisfying some peoples craving to

freebsd-arch: Re: adding if_dev member to struct ifnet

have device_t in all data structures in the system.

Demanding such a relationship will only make our life more difficult when we get to deal with all the non-standard devices like if_sl, if_ppp, if_ng, if_tun, if_tap which have no device_t, or musycc, a network card where you have to juggle two device_t's, one for the framer and one for the line encoder.

As it is now, device_t and newbus provides a good model for our attachment of device drivers to hardware, it's not quite perfect, but it is good enough that nobody can be bothered to sit down and write something perfect.

We have "struct ifnet" which describes a network interface, it describes it based on the access model, and it does in fact not care a hoot what implements that interface, hardware, software or bongo drums, it doesn't matter.

Similarly, we have "dev_t" to describe filesystem accessed devices, and it describes those in the terms of the access model, not in terms of what is behind them, hardware, software or bongo drums.

If all you want is an extra field in "struct ifnet" to hang driver information on, then by all means add that field. As long as you give it type "void *" and make it private to the driver I have no problem with that.

--

Poul-Henning Kamp		UNIX since Zilog Zeus 3.20
phk@FreeBSD.ORG		TCP/IP since RFC 956
FreeBSD committer		BSD since 4.3-tahoe

Never attribute to malice what can adequately be explained by incompetence.

freebsd-arch@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-arch>

To unsubscribe, send any mail to "freebsd-arch-unsubscribe@freebsd.org"