

## Re: cvs commit: src/sys/modules/iwi Makefile src/sys/dev/iwi if\_iwi.c if\_iwireg.h if\_iwivar.h

*Source:* <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2005-11/0081.html>

---

*From:* Warner Losh (*imp\_at\_bsdimp.com*)

*Date:* 11/21/05

Date: Mon, 21 Nov 2005 11:19:55 -0700 (MST)

To: nate@root.org

jhb>John Baldwin

jhb> The easiest way to accomplish this is for your driver to send a  
jhb> message to devd requesting that the firmware be reloaded on  
jhb> resume since devd won't run until the kernel is fully back up.

des>Dag-Erling Smørgrav

des> ...or keep the firmware image in memory after loading it the first  
des> time around.

scottl>Scott Long

scottl> Or have the firmware be embedded in a KLD, like ispfw.

nate>Nate Lawson

nate> I think I've now been repeated by everyone in the conversation. :)

Only those people that think it is a good idea have repeated it.

nate> So maybe it's time to solve this? Move discussion to arch@?

I don't like the kld option. People have been talking about doing  
this since about 4.2 and nothing has happened to make it generic.

The good things about it are that the driver can request that modules  
be loaded and unloaded. Once loaded, the driver can go directly to a  
binary representation of the firmware. These are both good points.

The bad points about this is that you have to generate the firmware  
kld module. This will require a per-driver program to parse the  
firmware and some design to place the data into data structures that  
the driver can use to bang the data into the card. It would mean  
creating two copies of the firmware because most people will install  
the new firmware, then run this program and install the new kld (maybe  
the kld generation program would run each installkernel, since you  
never know what it depends on and when that might change). Also, you  
have the potential for version skew between the kld and the firmware.

freebsd-arch: Re: cvs commit: src/sys/modules/iwi Makefile src/sys/dev/iwi if\_iwi.c if\_iwireg.h if\_iwivar.h

There's one more level of indirection that you need to worry about when you go the kld route.

The firmware parsing code tends to be relatively simple and may need access to multiple files, as well as the actual hardware. The wi firmware loader, for example, needs to patch certain values from the card into the wi images before they will work (the current hand-tweaked symbol CF card has had these re-applied).

To solve the race with "/", one could easily just queue the driver loading to a taskqueue. I don't think that the taskqueues aren't running until after the resume process is complete. Even if they are, and you block waiting for / to come back, it wouldn't be the resume path that's blocking. One could also use devd for this, but if the driver already knows how to load its firmware, that seems like an overly complicated solution.

Warner

---

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"