

Re: Porting OpenBSD's sysctl hw.sensors framework to FreeBSD

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2007-07/msg00053.html>

- *From:* Alexander Leidinger <Alexander@xxxxxxxxxxxxxx>
 - *Date:* Wed, 11 Jul 2007 13:49:59 +0200
-

Quoting Robert Watson <rwatson@xxxxxxxxxxxxxx> (from Wed, 11 Jul 2007 11:12:24 +0100 (BST)):

On Wed, 11 Jul 2007, Poul-Henning Kamp wrote:

In message <469420B9.20401@xxxxxxxxxxxxxx>, "Constantine A. Murenin" writes:

If you want to have no such framework that could potentially diagnose or predict system failure, it's your choice, [...]

I would love to have that, but the OpenBSD code isn't that.

In the general spirit of SoC, I would suggest that a more constructive line of commenting might come with suggestions, not just rejections :-). Are you arguing that the current proposed framework offers little incremental benefit over simply having the sysctl framework in the first place and having each source of information (i.e., device driver) just export it directly?

It seems clear that people would like all these measurements to be available, even if not by the precise mechanism proposed. So far the specific technical criticals have been:

– There's such a diversity of motherboard devices and probe mechanisms that any kernel driver would become rapidly over-burdened and needlessly complicated.

This doesn't argue for doing nothing, just that perhaps a kernel device driver is the wrong place.

On the other hand you don't want to allow an userland tool to directly mess around with the registers on your RAID or NIC to get some status...

Re: Porting OpenBSD's sysctl hw.sensors framework to FreeBSD

This SoC project is not about writing a driver for the sensors, it's about importing a framework which provides interfaces to get sensor information and uses some standardized driver interfaces to get this information. At least as far as I have understand it (I haven't looked at the code). So I think about it as something which allows to add some code to e.g. the ATA driver which registers itself with the sensors framework. The user doesn't has to learn how to query the ATA stats when he already knows how to use the userland part of the sensors framework (like we have with ifconfig for network cards), and the driver writers don't have to invent the wheel again and again (like we have with cam or maybe in some way the NIC phys).

What Poul-Henning is talking about looks to me like a presentation layer, while I see the sensors framework (as far as I understand it) as an infrastructure layer. If not, please be more verbose Poul-Henning. It may be the case that the presentation layer needs some more meta information about a sensor which maybe the driver has to provide in some way, but without an infrastructure layer we don't get to the presentation layer.

Bye,
Alexander.

—

We may not return the affection of those who like us,
but we always respect their good judgement.

<http://www.Leidinger.net> Alexander @ Leidinger.net: PGP ID = B0063FE7
<http://www.FreeBSD.org> netchild @ FreeBSD.org : PGP ID = 72077137

freebsd-arch@xxxxxxxxxxx mailing list

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

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