

Re: Generic Kernel API

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/current/2005-11/0371.html>

From: Marcin Jessa (lists_at_yazzy.org)

Date: 11/09/05

Date: Wed, 9 Nov 2005 09:35:52 +0100

To: Scott Long <scottl@samsco.org>

On Tue, 08 Nov 2005 17:42:07 -0700

Scott Long <scottl@samsco.org> wrote:

> *Marcin Jessa wrote:*

> > *Hi guys.*

> >

> > *I just came across an article of Mr. Greg Kroah-Hartman in his blog*

> > <http://www.kroah.com/log/2005/11/03/>

> > *about generic kernel API which could make it possible for hardware*

> > *vendors to supply us with their own drivers.*

> > *To be honest I disagree with Greg and consider this a good idea.*

> > *Especially if we had a system which could isolate each device driver*

> > *running as a separate user-mode process so it would not bring down*

> > *the entire OS in case the driver was buggy.*

> > *An API like that would both boost up FreeBSD's popularity and make*

> > *it possible to use a way larger variety of hardware.*

> > *I mean, lets not fool ourselves, the majority of hardware vendors is*

> > *not interested in revealing of their secrets publishing freely*

> > *available documentation of their products.*

> > *We could have a new choice to use (or not) binary drivers the*

> > *same way the popular commercial O.Ss do.*

> > *What do you guys think? What is the view of the*

> > *FreeBSD community on this metter?*

> > *Could this be concerned as a good idea ?*

> >

> >

> > *Cheers,*

> > *Marcin Jessa.*

>

>

> *Please don't take this the wrong way, but google for 'Universal Driver*

> *Interface'. Yes, this topic comes up every few years. It sounds*

> *like a good idea, but every OS has unique and incompatible ways of*

> *doing things.*

You've misunderstood me Scott.

I never meant it to be an universal API but a FreeBSD one only.

freebsd-current: Re: Generic Kernel API

I understand an universal cross-platform driver API would me a nearly impossible project.

My idea is to create an API for binary vendor drivers to make it easier for hardware vendors to create FreeBSD drivers the same way they can do for Windows or Mac OS X.

- > *Sure, it's easy to map malloc in FreeBSD to kmalloc in*
- > *Linux, but how do you map ithreads in FreeBSD and Solaris to Linux?*
- > *How do you map busdma in FreeBSD to busdma in NetBSD, let alone Linux*
- > *where there is little concept of a DMA abstraction? How do you map*
- > *newbus in FreeBSD to, um, nothing in Linux? How do you map VFS on*
- > *FreeBSD to VFS on Linux or Solaris? All of these things make such a*
- > *unification effort impossible to do without watering it down to where*
- > *it is either functionally useless or too slow and abstract to*
- > *matter. Ironically, Project Evil has bridged the gap the best, but*
- > *it limits its scope to the Windows NDIS API. It might be possible to*
- > *expand it to cover StorPort also, but forget about much more than*
- > *that.*

freebsd-current@freebsd.org mailing list

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

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