

Re: kldunload DIAGNOSTIC idea...

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2004-07/0079.html>

From: Scott Long (scottl_at_samsco.org)

Date: 07/20/04

Date: Tue, 20 Jul 2004 13:10:29 -0600

To: Brian Fundakowski Feldman <green@freebsd.org>

Brian Fundakowski Feldman wrote:

> *On Tue, Jul 20, 2004 at 08:39:57PM +0200, Poul-Henning Kamp wrote:*

>

>>*In message <20040720183213.GC1009@green.homeunix.org>, Brian Fundakowski Feldma*

>>*n writes:*

>>

>>>*On Tue, Jul 20, 2004 at 08:20:23PM +0200, Poul-Henning Kamp wrote:*

>>>

>>>>*I'm pulling hair out trying to make it guaranteed safe to unload device*

>>>>*driver modules, and the major pain here is to make sure there is no*

>>>>*thread stuck somewhere inside the code.*

>>>>

>>>>*That gave me the idea for a simple little DIAGNOSTIC check for kldunload:*

>>>>*run through the proc/thread table and look for any thread with an*

>>>>*instruction counter inside the range of pages we are going to unload.*

>>>>

>>>>*Any takers ?*

>>>>

>>>>*You mean any thread with a stack trace that includes an instruction*

>>>>*counter inside those pages, don't you?*

>>>>

>>>>*That would require us to unwind the stack which I think is overkill*

>>>>*for the purpose.*

>>>>

>>>>*The most likely case is that the thread is sleeping on something*

>>>>*inside the kld so just checking the instruction pointer would be*

>>>>*fine.*

>>>>

>>>>*Looking for sleep addresses inside the module might make sense too.*

>>>>

>>>>

>>>>*It's probably not overkill -- at least in my experience most of the*

>>>>*time a driver is "doing something" it is sleeping, so the address*

>>>>*will be in mi_switch() or somewhere way out there. Sleep addresses*

>>>>*on dynamic data addresses are also a lot more common than sleep*

>>>>*addresses on static/code addresses. If someone is interested in*

>>>>*doign this, it would be very informative, especially if it could*

freebsd-arch: Re: kldunload DIAGNOSTIC idea...

> *catch sleeps, pending timeouts, pending callouts, etc.*

>

busdma callbacks, cam callbacks, netisr callbacks, and on and on and on.

Scott

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"