

## Re: kldunload DIAGNOSTIC idea...

**Source:** <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/arch/2004-07/0080.html>

---

**From:** David Schultz (*das\_at\_FreeBSD.ORG*)

**Date:** 07/20/04

Date: Tue, 20 Jul 2004 13:37:39 -0700  
To: Poul-Henning Kamp <phk@phk.freebsd.dk>

On Tue, Jul 20, 2004, 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.*

But this is just a heuristic that may sometimes fail. The module might be holding resources or locks, it could have callbacks, etc. If we're going to offer a forcible unload option, I think it's crucial that it be safe. That is, a system administrator must be able to use it on a production system and be confident that it will not cause anything bad to happen. Otherwise, it doesn't really buy you much.

Also, doing this right (i.e. the hard way) may have other advantages. For instance, when any old module or driver fails an

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

internal consistency check right now, or in some cases, simply  
fails to allocate memory, the best course of action