Subject: Re: [PATCH] usbatm: Update to use the kthread api.
Posted by Duncan Sands on Wed, 03 Jan 2007 19:12:34 GMT
View Forum Message <> Reply to Message

On Friday 15 December 2006 11:45, Eric W. Biederman wrote:
> Christoph Hellwig <hch@infradead.org> writes:
>
> > On Fri, Dec 15, 2006 at 03:17:57AM -0700, Eric W. Biederman wrote:
> >
> > I think the right fix is to encehance the kthread infrastructure to
> > gracefully handle the case where the threat has stopped by itself
> > and doesn't exist anymore at the time where we call kthread_stop.
>
> Yep that is about where I thought we were at. Now we need to figure out
> how to do that cleanly, and sanely.

There's a completely different solution, which is to use a workqueue
instead of a kthread, with users providing a cancellation method.
Recall that the functionality is provided by usbatm to drivers:
they use it to perform slow initialization that is too slow to be
done in the probe method. They register with usbatm, providing a
"heavy_init" method. They could also provide a "heavy_cancel" method.
(Any special data that heavy_cancel needs can be stored in the
existing driver private data structure; this structure is already
passed to heavy_init). In the case of the speedtch driver, it
could place a completion in its private data structure; heavy_cancel
would just complete the completion. Rather than doing interruptible
sleeps, it can use wait_for_completion_timeout.

The only thing that worries me about this solution is... that you
can't shoot down firmware loading from userspace anymore. For
example, if heavy_init is blocked loading firmware when the system
is halted, it presumably won't react to the kill signal. Perhaps
it is unimportant; and if not, I guess I can just re-enable signals
in heavy_init.

Ciao,

Duncan.

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers