Subject: Re: [PATCH] usbatm: Update to use the kthread api.
Posted by Christoph Hellwig on Fri, 15 Dec 2006 10:35:01 GMT
View Forum Message <> Reply to Message

On Fri, Dec 15, 2006 at 03:17:57AM -0700, Eric W. Biederman wrote:

> While converting the usbatm driver we hit what is at least a partial snag.
> I was hoping to remove the sending of signals along with the rest of
> the conversion, but I hit a surprising use.
>
> The usb atm drivers have some long running initializers (several seconds
> potentially.   So the infrastructure forks off a kernel thread to run them.
>
> The code really does not care if the thread completes or does anything
> else until a usb disconnect comes in. The in wants to wait suggest the
> initialization code stop early and abort and then wait until the
> initialization is done.
>
> The practical problem is what is the best way to handle that case.
>
> Can we use the kthread_should_stop() test in a thread that can
> exit on it's own before kthread_stop is called?

Right now it can't.

> Are signals the best available mechanism to request that a thread
> stop that can exit on it's own.

Defintly not. signals should be avoided in kernel threads at all
cost.

> If we don't suggest to the thread to stop having it call
> complete_and_exit seems to the simplest race free solution. The
> request to stop though makes things trickier.

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.

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