Subject: Re: [PATCH] kthread: Spontaneous exit support
Posted by Christoph Hellwig on Mon, 23 Apr 2007 18:09:18 GMT
View Forum Message <> Reply to Message

On Mon, Apr 23, 2007 at 11:45:51AM -0600, Eric W. Biederman wrote:
> Ok. Thinking about it I agree with Christoph that parallel API's can
> be a problem.
>
> However we do still need to support kernel threads where kthread_stop will
> never be called. There appear to be a few legitimate cases where
> someone wants to fire off a thread and have it do some work but don't
> care at all for stopping it before it is done.

There's two cases where it's valid that kthread_stop is not called:

 a) the user is always builtin and the thread runs until the kernel halts.
    examples: voyager, arm ecard
 b) the thread is normally started/stopped, e.g. at module_init/module_exit
    but there is some reason why it could terminate earlier.
    examples: the various bluetooth threads, nfs-related threads that
    can be killed using signals
 c) we have some kind of asynchronous helper thread.
    examples: various s390 drivers, usbatm, therm_pm72
 d) a driver has threadpools were we need to start/stop threads on demand.
    examples: nfsd, xpc

case a)
 is trivial, we can just ignore the refcounting issue.

case b)
 is what refcounting the task struct and proper handling in
 kthread_stop will deal with.

case c)
 should get a new kthread_create_async api which starts a thread
 without blocking, so we can get rid of the workqueues in the
 s390 drivers. it should probably also be safe to be called from
 irq context. What makes this a bit complicated is the need to
 make sure no more thread is running in case the caller terminates
 (shutdown of the structure it's associated with or module removal)

case d)
 should be deal with with a kthread_pool api

_____
Containers mailing list
Containers@lists.linux-foundation.org