Subject: Re: [PATCH] mtd_blkdevs: Convert to use the kthread API
Posted by akpm on Thu, 19 Apr 2007 22:26:39 GMT
View Forum Message <> Reply to Message

On Thu, 19 Apr 2007 13:13:22 -0600
ebiederm@xmission.com (Eric W. Biederman) wrote:

> Christoph Hellwig <hch@infradead.org> writes:
>
> > On Thu, Apr 19, 2007 at 12:55:28AM -0600, Eric W. Biederman wrote:
> >> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
> >>
> >> thread_run is used intead of kernel_thread, daemonize, and mucking
> >> around blocking signals directly.
> >
> > Please don't do incomplete transitions like that. We don't really
> > want people to use kthread_run, but not the kthread stopping
> > mechanisms, because people will simply forget about that bit and
> > we'll never get rid of the enormous amount of, erm creativity, in
> > handling kernel thread stopping.
> >
> > This is just the first patch in your series where the thread is mutable,
> > but it equally applies to all following patches where this is the case
> > aswell.
>
> I don't really care about the creativity. Although it would
> be nice if it wasn't there. I deliberately left it in so I would be
> certain my patches were correct.
>
> I care about killing the maintenance and forward development roadblocks
> that are kernel_thread and daemonize. And the user interface problem
> that is handling signals in kernel threads.
>

Yes, I think that is a practical position, if not an ideal one.

MTD (to pick one example) does need to be decruftified: remove
r->blkcore_priv->exiting, probably ->blkcore_priv->thread_dead, switch
deregister_mtd_blktrans() to use kthread_stop(). But it's a bit much to
expect Eric to make that conversion, and to suitably test it. All he can
do is to make a best-effort and hope that someone else tests it, which
isn't very reliable.

This partial patch at least gets us some of the way there, and serves as a
gentle reminder to dwmwyouknowwho to finish cleaning this stuff up.

I'd be more concerned about a part-conversion in a subsystem which has no
identifiable maintainer, because in that case the chances are that we'll

just forget about it an the conversion would never be completed.

And of course, these are not simply cleanup patches: we actually need to get the kernel threads out of the daemonize() and signalling game to complete the virtualisation work.

_____