
Subject: [PATCH] mtd_blkdevs: Convert to use the kthread API

Posted by [ebiederm](#) on Thu, 19 Apr 2007 06:55:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

From: Eric W. Biederman <ebiederm@xmission.com> - unquoted

thread_run is used instead of kernel_thread, daemonize, and mucking around blocking signals directly.

CC: David Woodhouse <dwmw2@infradead.org>

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

drivers/mtd/mtd_blkdevs.c | 19 +++++-----

1 files changed, 5 insertions(+), 14 deletions(-)

```
diff --git a/drivers/mtd/mtd_blkdevs.c b/drivers/mtd/mtd_blkdevs.c
```

```
index db7397c..ed71d5e 100644
```

```
--- a/drivers/mtd/mtd_blkdevs.c
```

```
+++ b/drivers/mtd/mtd_blkdevs.c
```

```
@@ -21,6 +21,7 @@
```

```
#include <linux/init.h>
```

```
#include <linux/mutex.h>
```

```
#include <linux/freezer.h>
```

```
+#include <linux/kthread.h>
```

```
#include <asm/uaccess.h>
```

```
static LIST_HEAD(blktrans_majors);
```

```
@@ -84,17 +85,6 @@ static int mtd_blktrans_thread(void *arg)
```

```
/* we might get involved when memory gets low, so use PF_MEMALLOC */
```

```
current->flags |= PF_MEMALLOC | PF_NOFREEZE;
```

```
- daemonize("%sd", tr->name);
```

```
-
```

```
- /* daemonize() doesn't do this for us since some kernel threads
```

```
- actually want to deal with signals. We can't just call
```

```
- exit_sighand() since that'll cause an oops when we finally
```

```
- do exit. */
```

```
- spin_lock_irq(&current->sighand->siglock);
```

```
- sigfillset(&current->blocked);
```

```
- recalc_sigpending();
```

```
- spin_unlock_irq(&current->sighand->siglock);
```

```
-
```

```
spin_lock_irq(rq->queue_lock);
```

```
while (!tr->blkcore_priv->exiting) {
```

```
@@ -368,6 +358,7 @@ static struct mtd_notifier blktrans_notifier = {
```

```
int register_mtd_blktrans(struct mtd_blktrans_ops *tr)
```

```
{  
+ struct task_struct *task;  
int ret, i;  
  
/* Register the notifier if/when the first device type is  
@@ -406,13 +397,13 @@ int register_mtd_blktrans(struct mtd_blktrans_ops *tr)  
blk_queue_hardsect_size(tr->blkcore_priv->rq, tr->blksize);  
tr->blkshift = ffs(tr->blksize) - 1;  
  
- ret = kernel_thread(mtd_blktrans_thread, tr, CLONE_KERNEL);  
- if (ret < 0) {  
+ task = kthread_run(mtd_blktrans_thread, tr, "%sd", tr->name);  
+ if (IS_ERR(task)) {  
    blk_cleanup_queue(tr->blkcore_priv->rq);  
    unregister_blkdev(tr->major, tr->name);  
    kfree(tr->blkcore_priv);  
    mutex_unlock(&mtd_table_mutex);  
- return ret;  
+ return PTR_ERR(task);  
}  
  
INIT_LIST_HEAD(&tr->devs);  
--
```

1.5.0.g53756

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
