

---

Subject: [PATCH] mtd\_blkdevs: Convert to use the kthread API

Posted by [ebiederm](#) on Thu, 19 Apr 2007 07:58:33 GMT

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

---

From: Eric W. Biederman <[ebiederm@xmission.com](mailto:ebiederm@xmission.com)>

thread\_run is used instead of kernel\_thread, daemonize, and mucking around blocking signals directly.

CC: David Woodhouse <[dwmw2@infradead.org](mailto:dwmw2@infradead.org)>

Signed-off-by: Eric W. Biederman <[ebiederm@xmission.com](mailto: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>

---