

---

Subject: [PATCH] synchro\_test: Convert to the kthread API.  
Posted by [ebiederm](#) on Thu, 19 Apr 2007 07:59:06 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

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

Cc: David Howells <[dhowells@redhat.com](mailto:dhowells@redhat.com)>

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

---

kernel/synchro-test.c | 16 ++++++-----  
1 files changed, 6 insertions(+), 10 deletions(-)

diff --git a/kernel/synchro-test.c b/kernel/synchro-test.c

index a4747a6..b1d7fd6 100644

--- a/kernel/synchro-test.c

+++ b/kernel/synchro-test.c

@@ -30,6 +30,7 @@

#include <linux/timer.h>

#include <linux/completion.h>

#include <linux/mutex.h>

+#include <linux/kthread.h>

#define MAX\_THREADS 64

@@ -224,7 +225,6 @@ static int mutexer(void \*arg)

{  
 unsigned int N = (unsigned long) arg;

- daemonize("Mutex%u", N);

set\_user\_nice(current, 19);

while (atomic\_read(&do\_stuff)) {

@@ -246,7 +246,6 @@ static int semaphorer(void \*arg)

{  
 unsigned int N = (unsigned long) arg;

- daemonize("Sem%u", N);

set\_user\_nice(current, 19);

while (atomic\_read(&do\_stuff)) {

@@ -268,7 +267,6 @@ static int reader(void \*arg)

{  
 unsigned int N = (unsigned long) arg;

- daemonize("Read%u", N);

set\_user\_nice(current, 19);

while (atomic\_read(&do\_stuff)) {

```

@@ -292,7 +290,6 @@ static int writer(void *arg)
{
    unsigned int N = (unsigned long) arg;

- daemonize("Write%u", N);
  set_user_nice(current, 19);

    while (atomic_read(&do_stuff)) {
@@ -316,7 +313,6 @@ static int downgrader(void *arg)
{
    unsigned int N = (unsigned long) arg;

- daemonize("Down%u", N);
  set_user_nice(current, 19);

    while (atomic_read(&do_stuff)) {
@@ -433,27 +429,27 @@ static int __init do_tests(void)
for (loop = 0; loop < MAX_THREADS; loop++) {
    if (loop < nummx) {
        init_completion(&mx_comp[loop]);
- kernel_thread(mutexer, (void *) loop, 0);
+ kthread_run(mutexer, (void *) loop, "Mutex%u", loop);
    }

    if (loop < numsm) {
        init_completion(&sm_comp[loop]);
- kernel_thread(semaphorer, (void *) loop, 0);
+ kthread_run(semaphorer, (void *) loop, "Sem%u", loop);
    }

    if (loop < numrd) {
        init_completion(&rd_comp[loop]);
- kernel_thread(reader, (void *) loop, 0);
+ kthread_run(reader, (void *) loop, "Read%u", loop);
    }

    if (loop < numwr) {
        init_completion(&wr_comp[loop]);
- kernel_thread(writer, (void *) loop, 0);
+ kthread_run(writer, (void *) loop, "Write%u", loop);
    }

    if (loop < numdg) {
        init_completion(&dg_comp[loop]);
- kernel_thread(downgrader, (void *) loop, 0);
+ kthread_run(downgrader, (void *) loop, "Down%u", loop);
    }
}

```

--

1.5.0.g53756

---

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>

---