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

Posted by [ebiederm](#) on Thu, 19 Apr 2007 06:55:29 GMT

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

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

kthread_run replaces the kernel_thread and daemonize calls during thread startup.

Calls to signal_pending were also removed as it is currently impossible for the cpci_hotplug thread to receive signals.

CC: Scott Murray <scottm@somanetworks.com>

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

drivers/pci/hotplug/cpci_hotplug_core.c | 22 ++++++-----
1 files changed, 7 insertions(+), 15 deletions(-)

diff --git a/drivers/pci/hotplug/cpci_hotplug_core.c b/drivers/pci/hotplug/cpci_hotplug_core.c
index 6845515..c620c7e 100644

--- a/drivers/pci/hotplug/cpci_hotplug_core.c

+++ b/drivers/pci/hotplug/cpci_hotplug_core.c

@@ -33,6 +33,7 @@

#include <linux/init.h>

#include <linux/interrupt.h>

#include <linux/smp_lock.h>

+#include <linux/kthread.h>

#include <asm/atomic.h>

#include <linux/delay.h>

#include "cpci_hotplug.h"

@@ -521,17 +522,13 @@ event_thread(void *data)

{
 int rc;

- lock_kernel();

- daemonize("cpci_hp_eventd");

- unlock_kernel();

-

dbg("%s - event thread started", __FUNCTION__);

while (1) {

dbg("event thread sleeping");

down_interruptible(&event_semaphore);

dbg("event thread woken, thread_finished = %d",
 thread_finished);

- if (thread_finished || signal_pending(current))

+ if (thread_finished)

break;

do {

```

    rc = check_slots();
@@ -562,12 +559,8 @@ poll_thread(void *data)
{
    int rc;

- lock_kernel();
- daemonize("cpci_hp_polld");
- unlock_kernel();
-
    while (1) {
- if (thread_finished || signal_pending(current))
+ if (thread_finished)
        break;
        if (controller->ops->query_enum()) {
            do {
@@ -592,7 +585,7 @@ poll_thread(void *data)
static int
cpci_start_thread(void)
{
- int pid;
+ struct task_struct *task;

    /* initialize our semaphores */
    init_MUTEX_LOCKED(&event_semaphore);
@@ -600,14 +593,13 @@ cpci_start_thread(void)
    thread_finished = 0;

    if (controller->irq)
- pid = kernel_thread(event_thread, NULL, 0);
+ task = kthread_run(event_thread, NULL, "cpci_hp_eventd");
    else
- pid = kernel_thread(poll_thread, NULL, 0);
- if (pid < 0) {
+ task = kthread_run(poll_thread, NULL, "cpci_hp_polld");
+ if (IS_ERR(task)) {
        err("Can't start up our thread");
        return -1;
    }
- dbg("Our thread pid = %d", pid);
    return 0;
}

--
1.5.0.g53756

```

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

