Subject: [PATCH] macintosh/adb: Convert to the kthread API
Posted by ebiederm on Thu, 19 Apr 2007 06:55:44 GMT
View Forum Message <> Reply to Message

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

This patch modifies the startup of kadbprobe to use
kthread_run instead of scheduling a work event which
later calls kernel_thread and in the thread calls
daemonize and blocks signals. kthread_run is simpler
and more maintainable.

The variable pid_t adb_probe_task_pid is replaced by
a struct task_struct variable named adb_probe_task.
Which works equally well with for testing if the current
process is the adb_probe thread, does not get confused
in the presence of a pid namespace and is easier to
compare against current as it is the same type.

The result is code that is slightly simpler and easier
to maintain.

Cc: Benjamin Herrenschmidt <benh@kernel.crashing.org>
Cc: Paul Mackerras <paulus@samba.org>
Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 drivers/macintosh/adb.c | 32 +++++++-------------------------
 1 files changed, 7 insertions(+), 25 deletions(-)

```
diff --git a/drivers/macintosh/adb.c b/drivers/macintosh/adb.c
index adfea3c..09c5261 100644
--- a/drivers/macintosh/adb.c
+++ b/drivers/macintosh/adb.c
@@ -35,6 +35,7 @@
 #include <linux/spinlock.h>
 #include <linux/completion.h>
 #include <linux/device.h>
+#include <linux/kthread.h>

 #include <asm/uaccess.h>
 #include <asm/semaphore.h>
@@ -82,7 +83,7 @@ struct adb_driver *adb_controller;
 BLOCKING_NOTIFIER_HEAD(adb_client_list);
 static int adb_got_sleep;
 static int adb_inited;
-static pid_t adb_probe_task_pid;
+static struct task_struct *adb_probe_task;
 static DECLARE_MUTEX(adb_probe_mutex);
```

```
 static struct completion adb_probe_task_comp;
 static int sleepy_trackpad;
@@ -137,8 +138,7 @@ static void printADBreply(struct adb_request *req)

 static __inline__ void adb_wait_ms(unsigned int ms)
 {
- if (current->pid && adb_probe_task_pid &&
-   adb_probe_task_pid == current->pid)
+ if (adb_probe_task == current)
   msleep(ms);
  else
   mdelay(ms);
@@ -245,35 +245,19 @@ static int adb_scan_bus(void)
  * This kernel task handles ADB probing. It dies once probing is
  * completed.
  */
-static int
-adb_probe_task(void *x)
+static int adb_probe(void *x)
 {
- sigset_t blocked;
-
- strcpy(current->comm, "kadbprobe");
-
- sigfillset(&blocked);
- sigprocmask(SIG_BLOCK, &blocked, NULL);
- flush_signals(current);

  printk(KERN_INFO "adb: starting probe task...\n");
  do_adb_reset_bus();
  printk(KERN_INFO "adb: finished probe task...\n");

- adb_probe_task_pid = 0;
+ adb_probe_task = NULL;
  up(&adb_probe_mutex);

  return 0;
 }

-static void
-__adb_probe_task(struct work_struct *bullshit)
-{
- adb_probe_task_pid = kernel_thread(adb_probe_task, NULL, SIGCHLD | CLONE_KERNEL);
-}
-
-static DECLARE_WORK(adb_reset_work, __adb_probe_task);
-
 int
```

```
 adb_reset_bus(void)
 {
@@ -283,7 +267,7 @@ adb_reset_bus(void)
  }

  down(&adb_probe_mutex);
- schedule_work(&adb_reset_work);
+ adb_probe_task = kthread_run(adb_probe, NULL, "kadbprobe");
  return 0;
 }

@@ -469,9 +453,7 @@ adb_request(struct adb_request *req, void (*done)(struct adb_request *),
  /* Synchronous requests send from the probe thread cause it to
   * block. Beware that the "done" callback will be overriden !
   */
- if ((flags & ADBREQ_SYNC) &&
-     (current->pid && adb_probe_task_pid &&
-     adb_probe_task_pid == current->pid)) {
+ if ((flags & ADBREQ_SYNC) && (current == adb_probe_task)) {
   req->done = adb_probe_wakeup;
   rc = adb_controller->send_request(req, 0);
   if (rc || req->complete)
--
1.5.0.g53756
```

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