Subject: Re: [PATCH] i386 voyager: Convert the monitor thread to use the kthread API
Posted by Christoph Hellwig on Sun, 22 Apr 2007 19:30:43 GMT
View Forum Message <> Reply to Message

On Thu, Apr 19, 2007 at 12:55:27AM -0600, Eric W. Biederman wrote:
> From: Eric W. Biederman <ebiederm@xmission.com> - unquoted
>
> This patch just trivially replaces kernel_thread and daemonize
> with a single call to kthread_run.

Here's a better patch that does the full kthread conversion +
switch to wake_up_process. Only compile tested of course due to
lack of voyager hardware.

Signed-off-by: Christoph Hellwig <hch@lst.de>

Index: linux-2.6/arch/i386/mach-voyager/voyager_cat.c
===
--- linux-2.6.orig/arch/i386/mach-voyager/voyager_cat.c 2007-04-22 15:19:28.000000000 +0200
+++ linux-2.6/arch/i386/mach-voyager/voyager_cat.c 2007-04-22 15:27:03.000000000 +0200
@@ -1111,7 +1111,7 @@ voyager_cat_do_common_interrupt(void)
     printk(KERN_ERR "Voyager front panel switch turned off\n");
     voyager_status.switch_off = 1;
     voyager_status.request_from_kernel = 1;
-    up(&kvoyagerd_sem);
+   wake_up_process(voyager_thread);
    }
    /* Tell the hardware we're taking care of the
     * shutdown, otherwise it will power the box off
@@ -1157,7 +1157,7 @@ voyager_cat_do_common_interrupt(void)
    outb(VOYAGER_CAT_END, CAT_CMD);
    voyager_status.power_fail = 1;
    voyager_status.request_from_kernel = 1;
-   up(&kvoyagerd_sem);
+   wake_up_process(voyager_thread);
   }

Index: linux-2.6/arch/i386/mach-voyager/voyager_thread.c
===
--- linux-2.6.orig/arch/i386/mach-voyager/voyager_thread.c 2007-04-22 15:15:24.000000000 +0200
+++ linux-2.6/arch/i386/mach-voyager/voyager_thread.c 2007-04-22 15:25:51.000000000 +0200
@@ -24,33 +24,16 @@
 #include <linux/kmod.h>
 #include <linux/completion.h>

```
 #include <linux/sched.h>
+#include <linux/kthread.h>
 #include <asm/desc.h>
 #include <asm/voyager.h>
 #include <asm/vic.h>
 #include <asm/mtrr.h>
 #include <asm/msr.h>

-#define THREAD_NAME "kvoyagerd"

-/* external variables */
-int kvoyagerd_running = 0;
-DECLARE_MUTEX_LOCKED(kvoyagerd_sem);
-
-static int thread(void *);
-
-static __u8 set_timeout = 0;
-
-/* Start the machine monitor thread.  Return 1 if OK, 0 if fail */
-static int __init
-voyager_thread_start(void)
-{
- if(kernel_thread(thread, NULL, CLONE_KERNEL) < 0) {
-  /* This is serious, but not fatal */
-  printk(KERN_ERR "Voyager: Failed to create system monitor thread!!!\n");
-  return 1;
- }
- return 0;
-}
+struct task_struct *voyager_thread;
+static __u8 set_timeout;

 static int
 execute(const char *string)
@@ -110,31 +93,15 @@ check_continuing_condition(void)
  }
 }

-static void
-wakeup(unsigned long unused)
-{
- up(&kvoyagerd_sem);
-}
-
 static int
 thread(void *unused)
 {
- struct timer_list wakeup_timer;
```

```
-
- kvoyagerd_running = 1;
-
- daemonize(THREAD_NAME);
-
- set_timeout = 0;
-
- init_timer(&wakeup_timer);
-
- sigfillset(&current->blocked);
-
  printk(KERN_NOTICE "Voyager starting monitor thread\n");

- for(;;) {
-  down_interruptible(&kvoyagerd_sem);
+ for (;;) {
+  set_current_state(TASK_INTERRUPTIBLE);
+  schedule_timeout(set_timeout ? HZ : MAX_SCHEDULE_TIMEOUT);
+
   VDEBUG(("Voyager Daemon awoken\n"));
   if(voyager_status.request_from_kernel == 0) {
    /* probably awoken from timeout */
@@ -143,20 +110,26 @@ thread(void *unused)
    check_from_kernel();
    voyager_status.request_from_kernel = 0;
   }
-  if(set_timeout) {
-   del_timer(&wakeup_timer);
-   wakeup_timer.expires = HZ + jiffies;
-   wakeup_timer.function = wakeup;
-   add_timer(&wakeup_timer);
-  }
  }
 }

+static int __init
+voyager_thread_start(void)
+{
+ voyager_thread = kthread_run(thread, NULL, "kvoyagerd");
+ if (IS_ERR(voyager_thread)) {
+  printk(KERN_ERR "Voyager: Failed to create system monitor thread.\n");
+  return PTR_ERR(voyager_thread);
+ }
+ return 0;
+}
+
+
 static void __exit
```

```
 voyager_thread_stop(void)
 {
- /* FIXME: do nothing at the moment */
+ kthread_stop(voyager_thread);
 }

 module_init(voyager_thread_start);
-//module_exit(voyager_thread_stop);
+module_exit(voyager_thread_stop);
Index: linux-2.6/include/asm-i386/voyager.h
===================================================================
--- linux-2.6.orig/include/asm-i386/voyager.h 2007-04-22 15:18:39.000000000 +0200
+++ linux-2.6/include/asm-i386/voyager.h 2007-04-22 15:24:13.000000000 +0200
@@ -487,15 +487,11 @@ extern struct voyager_qic_cpi *voyager_q
 extern struct voyager_SUS *voyager_SUS;

 /* variables exported always */
+extern struct task_struct *voyager_thread;
 extern int voyager_level;
-extern int kvoyagerd_running;
-extern struct semaphore kvoyagerd_sem;
 extern struct voyager_status voyager_status;

-
-
 /* functions exported by the voyager and voyager_smp modules */
-
 extern int voyager_cat_readb(__u8 module, __u8 asic, int reg);
 extern void voyager_cat_init(void);
 extern void voyager_detect(struct voyager_bios_info *);
```