Subject: [PATCH] vt: Rework the console spawning variables.
Posted by ebiederm on Sun, 10 Sep 2006 04:21:57 GMT
View Forum Message <> Reply to Message

This patch is against 2.6.18-rc6-mm1 with:
vt-update-spawnpid-to-be-a-struct-pid_t-tidy and
vt-update-spawnpid-to-be-a-struct-pid_t reverted. My previous version
of this patch did not correct the style that was already in use, and
failed to add proper SMP locking.

This is such a rare path it took me a while to figure out how to test
this after soring out the locking.

This patch does several things.
- The variables used are moved into a structure and declared in vt_kern.h
- A spinlock is added so we don't have SMP races updating the values.
- Instead of raw pid_t value a struct_pid is used to guard against
  pid wrap around issues, if the daemon to spawn a new console dies.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 drivers/char/keyboard.c | 16 ++++++++++++----
 drivers/char/vt_ioctl.c | 9 ++++++---
 include/linux/vt_kern.h | 7 +++++++
 3 files changed, 25 insertions(+), 7 deletions(-)

diff --git a/drivers/char/keyboard.c b/drivers/char/keyboard.c
index 3e90aac..99fb070 100644
--- a/drivers/char/keyboard.c
+++ b/drivers/char/keyboard.c
@@ -108,7 +108,11 @@ const int NR_TYPES = ARRAY_SIZE(max_vals
 struct kbd_struct kbd_table[MAX_NR_CONSOLES];
 static struct kbd_struct *kbd = kbd_table;

-int spawnpid, spawnsig;
+struct vt_spawn_console vt_spawn_con = {
+ .lock = SPIN_LOCK_UNLOCKED,
+ .pid = NULL,
+ .sig = 0,
+};

 /*
  * Variables exported for vt.c
@@ -578,9 +582,13 @@ static void fn_compose(struct vc_data *v

 static void fn_spawn_con(struct vc_data *vc, struct pt_regs *regs)
 {
- if (spawnpid)

```
-  if (kill_proc(spawnpid, spawnsig, 1))
-   spawnpid = 0;
+ spin_lock(&vt_spawn_con.lock);
+ if (vt_spawn_con.pid)
+  if (kill_pid(vt_spawn_con.pid, vt_spawn_con.sig, 1)) {
+   put_pid(vt_spawn_con.pid);
+   vt_spawn_con.pid = NULL;
+  }
+ spin_unlock(&vt_spawn_con.lock);
 }

 static void fn_SAK(struct vc_data *vc, struct pt_regs *regs)
diff --git a/drivers/char/vt_ioctl.c b/drivers/char/vt_ioctl.c
index a53e382..dc408af 100644
--- a/drivers/char/vt_ioctl.c
+++ b/drivers/char/vt_ioctl.c
@@ -645,13 +645,16 @@ #endif
   */
  case KDSIGACCEPT:
  {
-  extern int spawnpid, spawnsig;
   if (!perm || !capable(CAP_KILL))
    return -EPERM;
   if (!valid_signal(arg) || arg < 1 || arg == SIGKILL)
    return -EINVAL;
-  spawnpid = current->pid;
-  spawnsig = arg;
+
+  spin_lock_irq(&vt_spawn_con.lock);
+  put_pid(vt_spawn_con.pid);
+  vt_spawn_con.pid = get_pid(task_pid(current));
+  vt_spawn_con.sig = arg;
+  spin_unlock_irq(&vt_spawn_con.lock);
   return 0;
  }

diff --git a/include/linux/vt_kern.h b/include/linux/vt_kern.h
index 1009d3f..37a1a41 100644
--- a/include/linux/vt_kern.h
+++ b/include/linux/vt_kern.h
@@ -84,4 +84,11 @@ #define CON_BUF_SIZE (CONFIG_BASE_SMALL
 extern char con_buf[CON_BUF_SIZE];
 extern struct semaphore con_buf_sem;

+struct vt_spawn_console {
+ spinlock_t lock;
+ struct pid *pid;
+ int sig;
```

```
+};
+extern struct vt_spawn_console vt_spawn_con;
+
 #endif /* _VT_KERN_H */
--
1.4.2.rc3.g7e18e-dirty
```

_____

Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers