Subject: [PATCH] vt: Make SAK run in process context.
Posted by ebiederm on Mon, 11 Dec 2006 13:07:03 GMT
View Forum Message <> Reply to Message

This defers SAK so we can use the normal console semaphore to order
operations.

This removes the xchg operations that I used to attempt to attmically
update struct pid, because of the strange locking used for SAK. With
SAK using the normal console semaphore nothing special is needed.

Cc: Oleg Nesterov <oleg@tv-sign.ru>
Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
drivers/char/keyboard.c | 11 +++--------
drivers/char/sysrq.c | 7 ++++---
drivers/char/vt_ioctl.c | 27 +++++++++++++++++++++++++--
include/linux/vt_kern.h | 1 +
4 files changed, 33 insertions(+), 13 deletions(-)

```
diff --git a/drivers/char/keyboard.c b/drivers/char/keyboard.c
index 7a6c1c0..bd6912d 100644
--- a/drivers/char/keyboard.c
+++ b/drivers/char/keyboard.c
@@ -595,15 +595,10 @@ static void fn_spawn_con(struct vc_data *vc)

 static void fn_SAK(struct vc_data *vc)
 {
- struct tty_struct *tty = vc->vc_tty;
+ static DECLARE_WORK(SAK_work, deferred_SAK, NULL);
+ SAK_work.data = &vc_cons[fg_console];

- /*
-  * SAK should also work in all raw modes and reset
-  * them properly.
-  */
- if (tty)
-  do_SAK(tty);
- reset_vc(vc);
+ schedule_work(&SAK_work);
 }

 static void fn_null(struct vc_data *vc)
diff --git a/drivers/char/sysrq.c b/drivers/char/sysrq.c
index 05810c8..b4f67a0 100644
--- a/drivers/char/sysrq.c
+++ b/drivers/char/sysrq.c
@@ -61,9 +61,10 @@ static struct sysrq_key_op sysrq_loglevel_op = {
```

```
 #ifdef CONFIG_VT
 static void sysrq_handle_SAK(int key, struct tty_struct *tty)
 {
- if (tty)
-  do_SAK(tty);
- reset_vc(vc_cons[fg_console].d);
+ static DECLARE_WORK(SAK_work, deferred_SAK, NULL);
+ SAK_work.data = &vc_cons[fg_console];
+
+ schedule_work(&SAK_work);
 }
 static struct sysrq_key_op sysrq_SAK_op = {
  .handler = sysrq_handle_SAK,
diff --git a/drivers/char/vt_ioctl.c b/drivers/char/vt_ioctl.c
index 311493e..5ecc71e 100644
--- a/drivers/char/vt_ioctl.c
+++ b/drivers/char/vt_ioctl.c
@@ -672,7 +672,8 @@ int vt_ioctl(struct tty_struct *tty, struct file * file,
   vc->vt_mode = tmp;
   /* the frsig is ignored, so we set it to 0 */
   vc->vt_mode.frsig = 0;
-  put_pid(xchg(&vc->vt_pid, get_pid(task_pid(current))));
+  put_pid(vc->vt_pid);
+  vc->vt_pid = get_pid(task_pid(current));
   /* no switch is required -- saw@shade.msu.ru */
   vc->vt_newvt = -1;
   release_console_sem();
@@ -1063,12 +1064,34 @@ void reset_vc(struct vc_data *vc)
  vc->vt_mode.relsig = 0;
  vc->vt_mode.acqsig = 0;
  vc->vt_mode.frsig = 0;
- put_pid(xchg(&vc->vt_pid, NULL));
+ put_pid(vc->vt_pid);
+ vc->vt_pid = NULL;
  vc->vt_newvt = -1;
  if (!in_interrupt())    /* Via keyboard.c:SAK() - akpm */
   reset_palette(vc);
 }

+void deferred_SAK(void *data)
+{
+ struct vc *vc_con = data;
+ struct vc_data *vc;
+ struct tty_struct *tty;
+
+ acquire_console_sem();
+ vc = vc_con->d;
+ if (vc) {
```

```
+  tty = vc->vc_tty;
+  /*
+   * SAK should also work in all raw modes and reset
+   * them properly.
+   */
+  if (tty)
+   do_SAK(tty);
+  reset_vc(vc);
+ }
+ release_console_sem();
+}
+
 /*
  * Performs the back end of a vt switch
  */
diff --git a/include/linux/vt_kern.h b/include/linux/vt_kern.h
index 37a1a41..a727896 100644
--- a/include/linux/vt_kern.h
+++ b/include/linux/vt_kern.h
@@ -74,6 +74,7 @@ int con_copy_unimap(struct vc_data *dst_vc, struct vc_data *src_vc);
 int vt_waitactive(int vt);
 void change_console(struct vc_data *new_vc);
 void reset_vc(struct vc_data *vc);
+void deferred_SAK(void *data);

 /*
  * vc_screen.c shares this temporary buffer with the console write code so that
--
1.4.4.1.g278f
```

_____

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