Subject: [PATCH 3/20] Move checking for init's death in do_exit()
Posted by Pavel Emelianov on Tue, 07 Aug 2007 09:29:40 GMT
View Forum Message <> Reply to Message

From: Oleg Nesterov <oleg@tv-sign.ru>

If some system uses multi-threaded init the kernel panics when the main thread exits. Perhaps this is just a "quality of implementation" question. sys_exit() from the main thread should be OK, why /sbin/init should be special? That said, I personally do not think that multi-threaded init is terribly useful.

So I think the patch below makes sense for now. Note that it removes the games with pid_ns->child_reaper: this doesn't work currently, and this has to be modified when we actually support pid namespaces anyway.

Signed-off-by: Oleg Nesterov <oleg@tv-sign.ru>
Acked-by: Pavel Emelyanov <xemul@openvz.org>

```
 exit.c |   17 +++++++++--------
 1 files changed, 9 insertions(+), 8 deletions(-)

--- ./kernel/exit.c.ve3 2007-08-06 12:14:53.000000000 +0400
+++ ./kernel/exit.c 2007-08-06 12:15:19.000000000 +0400
@@ -895,6 +895,14 @@ static void check_stack_usage(void)
 static inline void check_stack_usage(void) {}
 #endif

+static inline void exit_child_reaper(struct task_struct *tsk)
+{
+ if (likely(tsk->group_leader != task_child_reaper(tsk)))
+  return;
+
+ panic("Attempted to kill init!");
+}
+
 fastcall NORET_TYPE void do_exit(long code)
 {
  struct task_struct *tsk = current;
@@ -908,14 +916,6 @@ fastcall NORET_TYPE void do_exit(long co
    panic("Aiee, killing interrupt handler!");
  if (unlikely(!tsk->pid))
    panic("Attempted to kill the idle task!");
- if (unlikely(tsk == task_child_reaper(tsk))) {
-  if (task_active_pid_ns(tsk) != &init_pid_ns)
-   task_active_pid_ns(tsk)->child_reaper =
```

```
-     init_pid_ns.child_reaper;
-  else
-    panic("Attempted to kill init!");
- }
-

   if (unlikely(current->ptrace & PT_TRACE_EXIT)) {
    current->ptrace_message = code;
@@ -965,6 +965,7 @@ fastcall NORET_TYPE void do_exit(long co
   }
   group_dead = atomic_dec_and_test(&tsk->signal->live);
   if (group_dead) {
+   exit_child_reaper(tsk);
    hrtimer_cancel(&tsk->signal->real_timer);
    exit_itimers(tsk->signal);
   }
```