Subject: [PATCH] Fix the sys_setpgrp() to work between namespaces
Posted by Pavel Emelianov on Fri, 24 Aug 2007 08:47:31 GMT
View Forum Message <> Reply to Message

The check if (task_pgrp_nr(p) != pgid) is almost always true,
because pgid is a "virtual" pid and it is most often much
smaller than the "real" pgrp id of any task (because pids are
generated sequentially most of the time). This leads to the
task's pgrp is always reset, even if it is not needed.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

---

 kernel/sys.c | 12 +++++++-----
 1 files changed, 7 insertions(+), 5 deletions(-)

```
diff --git a/kernel/sys.c b/kernel/sys.c
index c7c4fa4..c827186 100644
--- a/kernel/sys.c
+++ b/kernel/sys.c
@@ -918,6 +918,7 @@ asmlinkage long sys_setpgid(pid_t pid, p
  struct task_struct *p;
  struct task_struct *group_leader = current->group_leader;
  int err = -EINVAL;
+ struct pid_namespace *ns;

  if (!pid)
   pid = task_pid_vnr(group_leader);
@@ -929,10 +930,12 @@ asmlinkage long sys_setpgid(pid_t pid, p
  /* From this point forward we keep holding onto the tasklist lock
   * so that our parent does not change from under us. -DaveM
   */
+ ns = current->nsproxy->pid_ns;
+
  write_lock_irq(&tasklist_lock);

  err = -ESRCH;
- p = find_task_by_pid_ns(pid, current->nsproxy->pid_ns);
+ p = find_task_by_pid_ns(pid, ns);
  if (!p)
   goto out;

@@ -958,10 +961,9 @@ asmlinkage long sys_setpgid(pid_t pid, p
   goto out;

  if (pgid != pid) {
-  struct task_struct *g =
```

```
-   find_task_by_pid_type_ns(PIDTYPE_PGID, pgid,
-     current->nsproxy->pid_ns);
+  struct task_struct *g;

+  g = find_task_by_pid_type_ns(PIDTYPE_PGID, pgid, ns);
    if (!g || task_session(g) != task_session(group_leader))
     goto out;
   }
@@ -970,7 +972,7 @@ asmlinkage long sys_setpgid(pid_t pid, p
   if (err)
    goto out;

- if (task_pgrp_nr(p) != pgid) {
+ if (task_pgrp_nr_ns(p, ns) != pgid) {
    struct pid *pid;

    detach_pid(p, PIDTYPE_PGID);
```