Subject: [PATCH 2/16] Miscelaneous preparations for namespaces
Posted by Pavel Emelianov on Fri, 06 Jul 2007 08:03:54 GMT
View Forum Message <> Reply to Message

The most important change is moving exit_task_namespaces()
inside exit_notify() to makes it possible to notify the
exiting task's parent. However this should be done before
release_task() to address the issue pointed by Sukadev with
NFS kernel thread.

Other changes are small and do not deserve separate description.

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

```
 include/linux/pid_namespace.h |    7 ++++---
 kernel/exit.c                 |    3 ++-
 kernel/pid.c                  |    2 ++
 3 files changed, 8 insertions(+), 4 deletions(-)

--- ./include/linux/pid_namespace.h.ve1 2007-07-06 10:58:57.000000000 +0400
+++ ./include/linux/pid_namespace.h 2007-07-06 11:03:18.000000000 +0400
@@ -4,7 +4,6 @@
 #include <linux/sched.h>
 #include <linux/mm.h>
 #include <linux/threads.h>
-#include <linux/pid.h>
 #include <linux/nsproxy.h>
 #include <linux/kref.h>

@@ -24,9 +23,10 @@ struct pid_namespace {

 extern struct pid_namespace init_pid_ns;

-static inline void get_pid_ns(struct pid_namespace *ns)
+static inline struct pid_namespace *get_pid_ns(struct pid_namespace *ns)
 {
  kref_get(&ns->kref);
+ return ns;
 }

 extern struct pid_namespace *copy_pid_ns(unsigned long flags, struct pid_namespace *ns);
@@ -39,7 +39,8 @@ static inline void put_pid_ns(struct pid

 static inline struct task_struct *child_reaper(struct task_struct *tsk)
 {
- return init_pid_ns.child_reaper;
```

```
+ BUG_ON(tsk != current);
+ return tsk->nsproxy->pid_ns->child_reaper;
 }

 #endif /* _LINUX_PID_NS_H */
--- ./kernel/exit.c.ve1 2007-07-06 11:02:55.000000000 +0400
+++ ./kernel/exit.c 2007-07-06 11:02:55.000000000 +0400
@@ -862,6 +862,8 @@ static void exit_notify(struct task_stru
   release_task(t);
  }

+ exit_task_namespaces(tsk);
+
  /* If the process is dead, release it - nobody will wait for it */
  if (state == EXIT_DEAD)
   release_task(tsk);
@@ -1002,7 +1004,6 @@ fastcall NORET_TYPE void do_exit(long co

  tsk->exit_code = code;
  proc_exit_connector(tsk);
- exit_task_namespaces(tsk);
  exit_notify(tsk);
 #ifdef CONFIG_NUMA
  mpol_free(tsk->mempolicy);
--- ./kernel/pid.c.ve1 2007-07-06 10:58:57.000000000 +0400
+++ ./kernel/pid.c 2007-07-06 11:02:55.000000000 +0400
@@ -71,6 +71,8 @@ struct pid_namespace init_pid_ns = {
  .child_reaper = &init_task
 };

+EXPORT_SYMBOL_GPL(init_pid_ns);
+
 /*
  * Note: disable interrupts while the pidmap_lock is held as an
  * interrupt might come in and do read_lock(&tasklist_lock).
```

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers