Subject: [PATCH 1/3] Introduce the dummy_pid
Posted by Pavel Emelianov on Wed, 03 Oct 2007 14:19:01 GMT
View Forum Message <> Reply to Message

This is a pid which is attached to tasks when they detach
their pids. This is done in detach_pid() and transfer_pid().
The pid_alive() check is changed to reflect this fact.

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

```
diff --git a/include/linux/sched.h b/include/linux/sched.h
index 4f21af1..e17b8f8 100644
--- a/include/linux/sched.h
+++ b/include/linux/sched.h
@@ -1286,9 +1286,11 @@ static inline pid_t task_ppid_nr_ns(stru
  * If pid_alive fails, then pointers within the task structure
  * can be stale and must not be dereferenced.
  */
+extern struct pid dummy_pid;
+
 static inline int pid_alive(struct task_struct *p)
 {
- return p->pids[PIDTYPE_PID].pid != NULL;
+ return p->pids[PIDTYPE_PID].pid != &dummy_pid;
 }

 /**
diff --git a/kernel/pid.c b/kernel/pid.c
index d7388d7..b7a11cf 100644
--- a/kernel/pid.c
+++ b/kernel/pid.c
@@ -81,6 +81,17 @@ struct pid_namespace init_pid_ns = {
 };
 EXPORT_SYMBOL_GPL(init_pid_ns);

+struct pid dummy_pid = {
+ .count   = ATOMIC_INIT(1),
+ .numbers = { {
+  .nr  = 0,  /* this is what pid_nr will return
+        * for tasks with no pids
+        */
+  .ns  = &init_pid_ns,
+ }, }
+};
+EXPORT_SYMBOL(dummy_pid);
+
```

```
 int is_cgroup_init(struct task_struct *tsk)
 {
  int ret = 0;
@@ -339,7 +350,7 @@ void fastcall detach_pid(struct task_str
  pid = link->pid;

  hlist_del_rcu(&link->node);
- link->pid = NULL;
+ link->pid = &dummy_pid;

  for (tmp = PIDTYPE_MAX; --tmp >= 0; )
   if (!hlist_empty(&pid->tasks[tmp]))
@@ -354,7 +365,7 @@ void fastcall transfer_pid(struct task_s
 {
  new->pids[type].pid = old->pids[type].pid;
  hlist_replace_rcu(&old->pids[type].node, &new->pids[type].node);
- old->pids[type].pid = NULL;
+ old->pids[type].pid = &dummy_pid;
 }

 struct task_struct * fastcall pid_task(struct pid *pid, enum pid_type type)
```