Subject: [RFC][PATCH 01/14] Define/use task_pid_ns() wrapper
Posted by Sukadev Bhattiprolu on Wed, 21 Mar 2007 03:11:21 GMT
View Forum Message <> Reply to Message

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Subject: [RFC][PATCH 01/14] Define/use task_pid_ns() wrapper

Use a wrapper to access/set pid_namespace of a task.

Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
---
 fs/exec.c                  |    2 +-
 fs/proc/proc_misc.c        |    2 +-
 include/linux/pid_namespace.h |    5 +++++
 kernel/exit.c              |    4 ++--
 kernel/pid.c               |    4 ++--
 5 files changed, 11 insertions(+), 6 deletions(-)

Index: 2.6.21-rc3-mm2/fs/exec.c
===================================================================
--- 2.6.21-rc3-mm2.orig/fs/exec.c
+++ 2.6.21-rc3-mm2/fs/exec.c
@@ -620,7 +620,7 @@ static int de_thread(struct task_struct
   * so it is safe to do it under read_lock.
   */
  if (unlikely(tsk->group_leader == child_reaper(tsk)))
-  tsk->nsproxy->pid_ns->child_reaper = tsk;
+  task_pid_ns(tsk)->child_reaper = tsk;

  zap_other_threads(tsk);
  read_unlock(&tasklist_lock);
Index: 2.6.21-rc3-mm2/fs/proc/proc_misc.c
===================================================================
--- 2.6.21-rc3-mm2.orig/fs/proc/proc_misc.c
+++ 2.6.21-rc3-mm2/fs/proc/proc_misc.c
@@ -93,7 +93,7 @@ static int loadavg_read_proc(char *page,
   LOAD_INT(a), LOAD_FRAC(a),
   LOAD_INT(b), LOAD_FRAC(b),
   LOAD_INT(c), LOAD_FRAC(c),
-  nr_running(), nr_threads, current->nsproxy->pid_ns->last_pid);
+  nr_running(), nr_threads, task_pid_ns(current)->last_pid);
  return proc_calc_metrics(page, start, off, count, eof, len);
 }

Index: 2.6.21-rc3-mm2/kernel/exit.c
===================================================================
--- 2.6.21-rc3-mm2.orig/kernel/exit.c
+++ 2.6.21-rc3-mm2/kernel/exit.c

```
@@ -790,8 +790,8 @@ fastcall NORET_TYPE void do_exit(long co
  if (unlikely(!tsk->pid))
   panic("Attempted to kill the idle task!");
  if (unlikely(tsk == child_reaper(tsk))) {
-  if (tsk->nsproxy->pid_ns != &init_pid_ns)
-   tsk->nsproxy->pid_ns->child_reaper = init_pid_ns.child_reaper;
+  if (task_pid_ns(tsk) != &init_pid_ns)
+   task_pid_ns(tsk)->child_reaper = init_pid_ns.child_reaper;
   else
    panic("Attempted to kill init!");
  }
Index: 2.6.21-rc3-mm2/kernel/pid.c
===================================================================
--- 2.6.21-rc3-mm2.orig/kernel/pid.c
+++ 2.6.21-rc3-mm2/kernel/pid.c
@@ -213,7 +213,7 @@ struct pid *alloc_pid(void)
  if (!pid)
   goto out;

- nr = alloc_pidmap(current->nsproxy->pid_ns);
+ nr = alloc_pidmap(task_pid_ns(current));
  if (nr < 0)
   goto out_free;

@@ -358,7 +358,7 @@ struct pid *find_ge_pid(int nr)
   pid = find_pid(nr);
   if (pid)
    break;
-  nr = next_pidmap(current->nsproxy->pid_ns, nr);
+  nr = next_pidmap(task_pid_ns(current), nr);
  } while (nr > 0);

  return pid;
Index: 2.6.21-rc3-mm2/include/linux/pid_namespace.h
===================================================================
--- 2.6.21-rc3-mm2.orig/include/linux/pid_namespace.h
+++ 2.6.21-rc3-mm2/include/linux/pid_namespace.h
@@ -37,6 +37,11 @@ static inline void put_pid_ns(struct pid
  kref_put(&ns->kref, free_pid_ns);
 }

+static inline struct pid_namespace *task_pid_ns(struct task_struct *tsk)
+{
+       return tsk->nsproxy->pid_ns;
+}
+
 static inline struct task_struct *child_reaper(struct task_struct *tsk)
 {
```

return init_pid_ns.child_reaper;

--

_____

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