Subject: [RFC][PATCH 03/14] use pid_nr in procfs
Posted by Sukadev Bhattiprolu on Wed, 21 Mar 2007 03:19:00 GMT
View Forum Message <> Reply to Message

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Subject: [RFC][PATCH 03/14] use pid_nr in procfs

With containers, a process can have different pid_t values in different
pid namespaces. To ensure we get the correct pid_t value in any context,
we should use pid_nr() function rather than directly accessing either
task->pid or pid->nr.

Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
---
 fs/proc/array.c | 16 +++++++++++------
 fs/proc/base.c | 4 ++--
 2 files changed, 12 insertions(+), 8 deletions(-)

Index: 2.6.21-rc3-mm2/fs/proc/array.c
==========
--- 2.6.21-rc3-mm2.orig/fs/proc/array.c
+++ 2.6.21-rc3-mm2/fs/proc/array.c
@@ -164,11 +164,15 @@ static inline char * task_state(struct t
  struct group_info *group_info;
  int g;
  struct fdtable *fdt = NULL;
+ pid_t ppid = 0;

  rcu_read_lock();
  tracer = tracehook_tracer_task(p);
  tracer_pid = tracer == NULL ? 0 : tracer->pid;

+ if (pid_alive(p))
+ ppid = pid_nr(task_tgid(rcu_dereference(p->parent)));
+
  buffer += sprintf(buffer,
   "State:\t%s\n"
   "SleepAVG:\t%lu%%\n"
@@ -180,8 +184,8 @@ static inline char * task_state(struct t
   "Gid:\t%d\t%d\t%d\t%d\n",
   get_task_state(p),
   (p->sleep_avg/1024)*100/(1020000000/1024),
-        p->tgid, p->pid,
-        pid_alive(p) ? rcu_dereference(p->parent)->tgid : 0,
+   pid_nr(task_tgid(p)), pid_nr(task_pid(p)),
+   ppid,
   tracer_pid,
   p->uid, p->euid, p->suid, p->fsuid,

```
   p->gid, p->egid, p->sgid, p->fsgid);
@@ -387,9 +391,9 @@ static int do_task_stat(struct task_stru
    stime = cputime_add(stime, sig->stime);
   }

-  sid = signal_session(sig);
-  pgid = process_group(task);
-  ppid = rcu_dereference(task->parent)->tgid;
+  sid = pid_nr(task_session(task));
+  pgid = pid_nr(task_pgrp(task));
+  ppid = pid_nr(task_tgid(rcu_dereference(task->parent)));

   unlock_task_sighand(task, &flags);
  }
@@ -419,7 +423,7 @@ static int do_task_stat(struct task_stru
  res = sprintf(buffer,"%d (%s) %c %d %d %d %d %d %u %lu \
 %lu %lu %lu %lu %lu %ld %ld %ld %ld %d 0 %llu %lu %ld %lu %lu %lu %lu %lu \
 %lu %lu %lu %lu %lu %lu %lu %lu %d %d %u %u %llu\n",
-  task->pid,
+  pid_nr(task_pid(task)),
   tcomm,
   state,
   ppid,
Index: 2.6.21-rc3-mm2/fs/proc/base.c
===================================================================
--- 2.6.21-rc3-mm2.orig/fs/proc/base.c
+++ 2.6.21-rc3-mm2/fs/proc/base.c
@@ -2132,7 +2132,7 @@ retry:
  task = NULL;
  pid = find_ge_pid(tgid);
  if (pid) {
-  tgid = pid->nr + 1;
+  tgid = pid_nr(pid) + 1;
   task = pid_task(pid, PIDTYPE_PID);
   /* What we to know is if the pid we have find is the
    * pid of a thread_group_leader.  Testing for task
@@ -2186,7 +2186,7 @@ int proc_pid_readdir(struct file * filp,
  for (task = next_tgid(tgid);
       task;
       put_task_struct(task), task = next_tgid(tgid + 1)) {
-  tgid = task->pid;
+  tgid = pid_nr(task_pid(task));
   filp->f_pos = tgid + TGID_OFFSET;
   if (proc_pid_fill_cache(filp, dirent, filldir, task, tgid) < 0) {
    put_task_struct(task);

--
```

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