
Subject: [PATCH 12/13] Show appropriate pids in proc
Posted by [Pavel Emelianov](#) on Thu, 24 May 2007 13:08:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

This is the proc-related part of the previous patch.
Since tasks are seen from two proc-s the appropriate
(virtual or global) pid must be shown.

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

```
diff --git a/fs/proc/array.c b/fs/proc/array.c
index aef7b7b..f65e4c9 100644
--- a/fs/proc/array.c
+++ b/fs/proc/array.c
@@ -75,6 +75,7 @@ 
 #include <linux/cpuset.h>
 #include <linux/rcupdate.h>
 #include <linux/delayacct.h>
+#include <linux/pid_namespace.h>

#include <asm/uaccess.h>
#include <asm/pgtable.h>
@@ -161,7 +162,9 @@ static inline char * task_state(struct t
 struct group_info *group_info;
 int g;
 struct fdtable *fdt = NULL;
+ struct pid_namespace *ns;

+ ns = current->nsproxy->pid_ns;
rcu_read_lock();
buffer += sprintf(buffer,
"State:\t%s\n"
@@ -174,9 +177,12 @@ 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->real_parent)->tgid : 0,
-     pid_alive(p) && p->ptrace ? rcu_dereference(p->parent)->pid : 0,
+     __task_tgid_nr_ns(p, ns),
+     __task_pid_nr_ns(p, ns),
+     pid_alive(p) ? task_ppid_nr_ns(p, ns) : 0,
+     pid_alive(p) && p->ptrace ?
+     __task_tgid_nr_ns(
+     rcu_dereference(p->parent), ns) : 0,
p->uid, p->euid, p->suid, p->fsuid,
```

```

p->gid, p->egid, p->sgid, p->fsgid);

@@ -349,6 +355,7 @@ static int do_task_stat(struct task_struct *task)
    rCU_read_lock();
    if (lock_task_sighand(task, &flags)) {
        struct signal_struct *sig = task->signal;
+       struct pid_namespace *ns = current->nsproxy->pid_ns;

        if (sig->tty) {
            tty_pgrp = pid_nr(sig->tty->pgrp);
@@ -381,9 +388,9 @@ static int do_task_stat(struct task_struct *task)
        stime = cputime_add(stime, sig->stime);
    }

-       sid = task_session_nr(task);
-       pgid = task_pgrp_nr(task);
-       ppid = rcu_dereference(task->real_parent)->tgid;
+       sid = __task_session_nr_ns(task, ns);
+       pgid = __task_pgrp_nr_ns(task, ns);
+       ppid = task_ppid_nr_ns(task, ns);

        unlock_task_sighand(task, &flags);
    }
@@ -414,7 +421,7 @@ static int do_task_stat(struct task_struct *task)
    res = sprintf(buffer,"%d (%s) %c %d %d %d %d %d %u %lu \
%lu %lu %lu %lu %lu %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,
+       task_pid_nr_ns(task),
        tcomm,
        state,
        ppid,
diff --git a/fs/proc/base.c b/fs/proc/base.c
index 8b426e9..e9811ec 100644
--- a/fs/proc/base.c
+++ b/fs/proc/base.c
@@ -73,6 +73,7 @@ 
#include <linux/poll.h>
#include <linux/nsproxy.h>
#include <linux/oom.h>
+#include <linux/pid_namespace.h>
#include "internal.h"

/* NOTE:
@@ -1755,14 +1756,14 @@ static int proc_self_readlink(struct dev_kern
    int buflen)
{
    char tmp[PROC_NUMBUF];

```

```
- sprintf(tmp, "%d", current->tgid);
+ sprintf(tmp, "%d", task_tgid_vnr(current));
    return vfs_readlink(dentry,buffer,buflen,tmp);
}

static void *proc_self_follow_link(struct dentry *dentry, struct nameidata *nd)
{
    char tmp[PROC_NUMBUF];
- sprintf(tmp, "%d", current->tgid);
+ sprintf(tmp, "%d", task_tgid_vnr(current));
    return ERR_PTR(vfs_follow_link(nd,tmp));
}
```
