
Subject: [PATCH 7/12] pid: Use struct pid for talking about process groups in exit.c
Posted by [ebiederm](#) on Wed, 13 Dec 2006 11:07:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

Modify has_stopped_jobs and will_become_orphan_pgrp to use struct pid based process groups. This reduces the number of hash tables looks ups and paves the way for multiple pid spaces.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

kernel/exit.c | 42 ++++++-----
1 files changed, 22 insertions(+), 20 deletions(-)

diff --git a/kernel/exit.c b/kernel/exit.c

index 97117e7..5f8455e 100644

--- a/kernel/exit.c

+++ b/kernel/exit.c

@@ -210,22 +210,22 @@ struct pid *session_of_pgrp(struct pid *pgrp)

*

* "I ask you, have you ever known what it is to be an orphan?"

*/

-static int will_become_orphaned_pgrp(int pgrp, struct task_struct *ignored_task)

+static int will_become_orphaned_pgrp(struct pid *pgrp, struct task_struct *ignored_task)

{

struct task_struct *p;

int ret = 1;

- do_each_task_pid(pgrp, PIDTYPE_PGID, p) {

+ do_each_pid_task(pgrp, PIDTYPE_PGID, p) {

if (p == ignored_task

|| p->exit_state

|| is_init(p->real_parent))

continue;

- if (process_group(p->real_parent) != pgrp &&

- process_session(p->real_parent) == process_session(p)) {

+ if (task_pgrp(p->real_parent) != pgrp &&

+ task_session(p->real_parent) == task_session(p)) {

ret = 0;

break;

}

- } while_each_task_pid(pgrp, PIDTYPE_PGID, p);

+ } while_each_pid_task(pgrp, PIDTYPE_PGID, p);

return ret; /* (sighing) "Often!" */

}

@@ -234,23 +234,23 @@ int is_orphaned_pgrp(int pgrp)

int retval;

```

    read_lock(&tasklist_lock);
-   retval = will_become_orphaned_pgrp(pgrp, NULL);
+   retval = will_become_orphaned_pgrp(find_pid(pgrp), NULL);
    read_unlock(&tasklist_lock);

```

```

    return retval;
}

```

```

-static int has_stopped_jobs(int pgrp)
+static int has_stopped_jobs(struct pid *pgrp)
{
    int retval = 0;
    struct task_struct *p;

-   do_each_task_pid(pgrp, PIDTYPE_PGID, p) {
+   do_each_pid_task(pgrp, PIDTYPE_PGID, p) {
        if (p->state != TASK_STOPPED)
            continue;
        retval = 1;
        break;
-   } while_each_task_pid(pgrp, PIDTYPE_PGID, p);
+   } while_each_pid_task(pgrp, PIDTYPE_PGID, p);
    return retval;
}

```

@@ -640,14 +640,14 @@ reparent_thread(struct task_struct *p, struct task_struct *father, int traced)

```

    * than we are, and it was the only connection
    * outside, so the child pgrp is now orphaned.
    */
-   if ((process_group(p) != process_group(father)) &&
-       (process_session(p) == process_session(father))) {
-   int pgrp = process_group(p);
+   if ((task_pgrp(p) != task_pgrp(father)) &&
+       (task_session(p) == task_session(father))) {
+   struct pid *pgrp = task_pgrp(p);

```

```

    if (will_become_orphaned_pgrp(pgrp, NULL) &&
        has_stopped_jobs(pgrp)) {
-   __kill_pg_info(SIGHUP, SEND_SIG_PRIV, pgrp);
-   __kill_pg_info(SIGCONT, SEND_SIG_PRIV, pgrp);
+   __kill_pgrp_info(SIGHUP, SEND_SIG_PRIV, pgrp);
+   __kill_pgrp_info(SIGCONT, SEND_SIG_PRIV, pgrp);
    }
}
}
}

```

@@ -727,6 +727,7 @@ static void exit_notify(struct task_struct *tsk)

```

int state;
struct task_struct *t;
struct list_head ptrace_dead, *_p, *_n;
+ struct pid *pgrp;

if (signal_pending(tsk) && !(tsk->signal->flags & SIGNAL_GROUP_EXIT)
    && !thread_group_empty(tsk)) {
@@ -779,12 +780,13 @@ static void exit_notify(struct task_struct *tsk)

    t = tsk->real_parent;

- if ((process_group(t) != process_group(tsk)) &&
-     (process_session(t) == process_session(tsk)) &&
-     will_become_orphaned_pgrp(process_group(tsk), tsk) &&
-     has_stopped_jobs(process_group(tsk))) {
-     __kill_pg_info(SIGHUP, SEND_SIG_PRIV, process_group(tsk));
-     __kill_pg_info(SIGCONT, SEND_SIG_PRIV, process_group(tsk));
+ pgrp = task_pgrp(tsk);
+ if ((task_pgrp(t) != pgrp) &&
+     (task_session(t) != task_session(tsk)) &&
+     will_become_orphaned_pgrp(pgrp, tsk) &&
+     has_stopped_jobs(pgrp)) {
+     __kill_pgrp_info(SIGHUP, SEND_SIG_PRIV, pgrp);
+     __kill_pgrp_info(SIGCONT, SEND_SIG_PRIV, pgrp);
+ }

    /* Let father know we died
--
1.4.4.1.g278f

```

Containers mailing list
Containers@lists.osdl.org
<https://lists.osdl.org/mailman/listinfo/containers>
