
Subject: [RFC][PATCH] Pass struct pid * to copy_process()
Posted by [Sukadev Bhattiprolu](#) on Tue, 16 Jan 2007 03:29:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

Modify copy_process() to take a struct pid * parameter instead of a pid_t.
This simplifies the code a bit and also avoids having to call find_pid()
to convert the pid_t to a struct pid.

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>

Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>

Cc: Cedric Le Goater <clg@fr.ibm.com>

Cc: Dave Hansen <haveblue@us.ibm.com>

Cc: Serge Hallyn <serue@us.ibm.com>

Cc: Eric Biederman <ebiederm@xmission.com>

Cc: containers@lists.osdl.org

kernel/fork.c | 46 ++++++-----
1 files changed, 23 insertions(+), 23 deletions(-)

Index: lx26-20-rc2-mm1/kernel/fork.c

```
=====
--- lx26-20-rc2-mm1.orig/kernel/fork.c 2007-01-15 18:05:50.079031592 -0800
+++ lx26-20-rc2-mm1/kernel/fork.c 2007-01-15 18:31:40.984258160 -0800
@@ -954,8 +954,9 @@ static struct task_struct *copy_process(
     unsigned long stack_size,
     int __user *parent_tidptr,
     int __user *child_tidptr,
-    int pid)
+    struct pid *spid)
 {
+ int nr = pid_nr(spid);
     int retval;
     struct task_struct *p = NULL;

@@ -1021,7 +1022,7 @@ static struct task_struct *copy_process(
     p->did_exec = 0;
     delayacct_tsk_init(p); /* Must remain after dup_task_struct() */
     copy_flags(clone_flags, p);
-    p->pid = pid;
+    p->pid = nr;
     retval = -EFAULT;
     if (clone_flags & CLONE_PARENT_SETTID)
         if (put_user(p->pid, parent_tidptr))
@@ -1239,27 +1240,25 @@ static struct task_struct *copy_process(
     }
 }
```

```

- if (likely(p->pid)) {
- add_parent(p);
- if (unlikely(p->ptrace & PT_PTRACED))
- __ptrace_link(p, current->parent);
-
- if (thread_group_leader(p)) {
- pid_t pgid = process_group(current);
- pid_t sid = process_session(current);
-
- p->signal->tty = current->signal->tty;
- p->signal->pgrp = pgid;
- set_signal_session(p->signal, sid);
- attach_pid(p, PIDTYPE_PGID, task_pgrp(current));
- attach_pid(p, PIDTYPE_SID, task_session(current));
+ add_parent(p);
+ if (unlikely(p->ptrace & PT_PTRACED))
+ __ptrace_link(p, current->parent);
+
+ if (thread_group_leader(p)) {
+ pid_t pgid = process_group(current);
+ pid_t sid = process_session(current);
+
+ p->signal->tty = current->signal->tty;
+ p->signal->pgrp = pgid;
+ set_signal_session(p->signal, sid);
+ attach_pid(p, PIDTYPE_PGID, task_pgrp(current));
+ attach_pid(p, PIDTYPE_SID, task_session(current));

- list_add_tail_rcu(&p->tasks, &init_task.tasks);
- __get_cpu_var(process_counts)++;
- }
- attach_pid(p, PIDTYPE_PID, find_pid(p->pid));
- nr_threads++;
+ list_add_tail_rcu(&p->tasks, &init_task.tasks);
+ __get_cpu_var(process_counts)++;
+ }
+ attach_pid(p, PIDTYPE_PID, spid);
+ nr_threads++;

total_forks++;
spin_unlock(&current->sigband->siglock);
@@ -1321,7 +1320,8 @@ struct task_struct * __devinit fork_idle
struct task_struct *task;
struct pt_regs regs;

- task = copy_process(CLONE_VM, 0, idle_regs(&regs), 0, NULL, NULL, 0);
+ task = copy_process(CLONE_VM, 0, idle_regs(&regs), 0, NULL, NULL,
+ &init_struct_pid);

```

```
if (!IS_ERR(task))
    init_idle(task, cpu);
```

```
@@ -1371,7 +1371,7 @@ long do_fork(unsigned long clone_flags,
    clone_flags |= CLONE_PTRACE;
}
```

```
- p = copy_process(clone_flags, stack_start, regs, stack_size, parent_tidptr, child_tidptr, nr);
+ p = copy_process(clone_flags, stack_start, regs, stack_size, parent_tidptr, child_tidptr, pid);
/*
 * Do this prior waking up the new thread - the thread pointer
 * might get invalid after that point, if the thread exits quickly.
```

Containers mailing list

Containers@lists.osdl.org

<https://lists.osdl.org/mailman/listinfo/containers>
