

---

Subject: [PATCH 03/17] Pid-NS(V3) Rename child\_reaper function.  
Posted by [Sukadev Bhattiprolu](#) on Sat, 16 Jun 2007 22:59:06 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

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

Rename the child\_reaper() function to task\_child\_reaper() to be similar to other task\_\* functions and to distinguish the function from 'struct pid\_namespace.child\_reaper'.

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

---  
fs/exec.c | 2 +-  
include/linux/pid\_namespace.h | 2 +-  
kernel/exit.c | 4 +++  
kernel/signal.c | 2 +-  
4 files changed, 5 insertions(+), 5 deletions(-)

Index: lx26-22-rc4-mm2/include/linux/pid\_namespace.h

=====

--- lx26-22-rc4-mm2.orig/include/linux/pid\_namespace.h 2007-06-13 16:54:55.000000000 -0700  
+++ lx26-22-rc4-mm2/include/linux/pid\_namespace.h 2007-06-13 16:55:50.000000000 -0700  
@@ -42,7 +42,7 @@ static inline struct pid\_namespace \*task  
return tsk->nsproxy->pid\_ns;  
}

-static inline struct task\_struct \*child\_reaper(struct task\_struct \*tsk)  
+static inline struct task\_struct \*task\_child\_reaper(struct task\_struct \*tsk)  
{  
return init\_pid\_ns.child\_reaper;  
}

Index: lx26-22-rc4-mm2/fs/exec.c

=====

--- lx26-22-rc4-mm2.orig/fs/exec.c 2007-06-13 16:54:55.000000000 -0700  
+++ lx26-22-rc4-mm2/fs/exec.c 2007-06-13 16:55:50.000000000 -0700  
@@ -631,7 +631,7 @@ static int de\_thread(struct task\_struct  
\* Reparenting needs write\_lock on tasklist\_lock,  
\* so it is safe to do it under read\_lock.  
\*/  
- if (unlikely(tsk->group\_leader == child\_reaper(tsk)))  
+ if (unlikely(tsk->group\_leader == task\_child\_reaper(tsk)))  
task\_active\_pid\_ns(tsk)->child\_reaper = tsk;

zap\_other\_threads(tsk);

Index: lx26-22-rc4-mm2/kernel/exit.c

=====

--- lx26-22-rc4-mm2.orig/kernel/exit.c 2007-06-13 16:54:55.000000000 -0700

+++ lx26-22-rc4-mm2/kernel/exit.c 2007-06-13 16:55:50.000000000 -0700

@ @ -695,7 +695,7 @ @ forget\_original\_parent(struct task\_struc

```
do {
    reaper = next_thread(reaper);
    if (reaper == father) {
-   reaper = child_reaper(father);
+   reaper = task_child_reaper(father);
        break;
    }
} while (reaper->exit_state);
```

@ @ -880,7 +880,7 @ @ fastcall NORET\_TYPE void do\_exit(long co

```
panic("Aiee, killing interrupt handler!");
if (unlikely(!tsk->pid))
    panic("Attempted to kill the idle task!");
- if (unlikely(tsk == child_reaper(tsk))) {
+ if (unlikely(tsk == task_child_reaper(tsk))) {
    if (task_active_pid_ns(tsk) != &init_pid_ns)
        task_active_pid_ns(tsk)->child_reaper =
            init_pid_ns.child_reaper;
```

Index: lx26-22-rc4-mm2/kernel/signal.c

=====

--- lx26-22-rc4-mm2.orig/kernel/signal.c 2007-06-13 16:42:12.000000000 -0700

+++ lx26-22-rc4-mm2/kernel/signal.c 2007-06-13 16:55:50.000000000 -0700

@ @ -1829,7 +1829,7 @ @ relock:

```
    * within that pid space. It can of course get signals from
    * its parent pid space.
    */
-   if (current == child_reaper(current))
+   if (current == task_child_reaper(current))
        continue;
```

```
    if (sig_kernel_stop(signr)) {
```

--

---

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>

---