

---

Subject: [PATCH 3/4] Clean up the kill\_something\_info  
Posted by [Pavel Emelianov](#) on Mon, 28 Jan 2008 15:16:35 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

This is the first stop (of two) in removing the kill\_pgrp\_info.

All the users of this function are in kernel/signal.c, but all they need is to call \_\_kill\_pgrp\_info() with the tasklist\_lock read-locked.

Fortunately, one of its users is the kill\_something\_info(), which already needs this lock in one of its branches, so clean these branches up and call the \_\_kill\_pgrp\_info() directly.

Based on Oleg's view of how this function should look.

Signed-off-by: Oleg Nesterov <oleg@tv-sign.ru>  
Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

---

```
kernel/signal.c | 26 ++++++-----  
1 files changed, 15 insertions(+), 11 deletions(-)
```

```
diff --git a/kernel/signal.c b/kernel/signal.c  
index 280bccb..ae9f6ba 100644  
--- a/kernel/signal.c  
+++ b/kernel/signal.c  
@@ -1133,14 +1133,22 @@ EXPORT_SYMBOL_GPL(kill_pid_info_as_uid);  
static int kill_something_info(int sig, struct siginfo *info, int pid)  
{  
    int ret;  
- rcu_read_lock();  
- if (!pid) {  
- ret = kill_pgrp_info(sig, info, task_pgrp(current));  
- } else if (pid == -1) {  
+  
+ if (pid > 0) {  
+ rcu_read_lock();  
+ ret = kill_pid_info(sig, info, find_vpid(pid));  
+ rcu_read_lock();  
+ return ret;  
+ }  
+  
+ read_lock(&tasklist_lock);  
+ if (pid != -1) {  
+ ret = __kill_pgrp_info(sig, info,  
+ pid ? find_vpid(-pid) : task_pgrp(current));
```

```

+ } else {
  int retval = 0, count = 0;
  struct task_struct * p;

- read_lock(&tasklist_lock);
  for_each_process(p) {
    if (p->pid > 1 && !same_thread_group(p, current)) {
      int err = group_send_sig_info(sig, info, p);
@@ -1149,14 +1157,10 @@ static int kill_something_info(int sig, struct siginfo *info, int pid)
      retval = err;
    }
  }
- read_unlock(&tasklist_lock);
  ret = count ? retval : -ESRCH;
- } else if (pid < 0) {
- ret = kill_pgrp_info(sig, info, find_vpid(-pid));
- } else {
- ret = kill_pid_info(sig, info, find_vpid(pid));
  }
- rcu_read_unlock();
+ read_unlock(&tasklist_lock);
+
  return ret;
}

```

--

1.5.3.4

---

Subject: Re: [PATCH 3/4] Clean up the kill\_something\_info

Posted by [akpm](#) on Mon, 04 Feb 2008 00:28:20 GMT

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

---

On Mon, 28 Jan 2008 18:16:35 +0300 Pavel Emelyanov <xemul@openvz.org> wrote:

```

> This is the first stop (of two) in removing the kill_pgrp_info.
>
> All the users of this function are in kernel/signal.c, but all
> they need is to call __kill_pgrp_info() with the tasklist_lock
> read-locked.
>
> Fortunately, one of its users is the kill_something_info(),
> which already needs this lock in one of its branches, so clean
> these branches up and call the __kill_pgrp_info() directly.
>
> Based on Oleg's view of how this function should look.

```

This patch causes my Fedora Core 3 x86\_64 machine to fail. When

rc.sys\_init runs /sbin/start\_udev a segmentation fault in start\_udev is reported and no device nodes are created. This is the only one of my test machines which behaves this way.

Config: <http://userweb.kernel.org/~akpm/config-x.txt>  
Running udev-039-8.FC3

---