
Subject: Re: [PATCH 14/15] Destroy pid namespace on init's death
Posted by [Oleg Nesterov](#) on Tue, 31 Jul 2007 09:07:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 07/30, sukadev@us.ibm.com wrote:

```
>
> --- lx26-23-rc1-mm1.orig/kernel/exit.c 2007-07-26 20:08:16.000000000 -0700
> +++ lx26-23-rc1-mm1/kernel/exit.c 2007-07-30 23:10:30.000000000 -0700
> @@ -915,6 +915,7 @@ fastcall NORET_TYPE void do_exit(long co
> {
>     struct task_struct *tsk = current;
>     int group_dead;
>     struct pid_namespace *pid_ns = tsk->nsproxy->pid_ns;
>
>     profile_task_exit(tsk);
>
> @@ -925,9 +926,10 @@ fastcall NORET_TYPE void do_exit(long co
>     if (unlikely(!tsk->pid))
>         panic("Attempted to kill the idle task!");
>     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;
> +     if (pid_ns != &init_pid_ns) {
> +         zap_pid_ns_processes(pid_ns);
> +         pid_ns->child_reaper = init_pid_ns.child_reaper;
> +     }
>     else
>         panic("Attempted to kill init!");
> }
```

Just to remind you, this is not right when init is multi-threaded,
we should do this only when the last thread exits.

```
> -static long do_wait(pid_t pid, int options, struct siginfo __user *infop,
> +long do_wait(pid_t pid, int options, struct siginfo __user *infop,
>     int __user *stat_addr, struct rusage __user *ru)
```

Small nit, other in-kernel reapers use sys_wait4(), perhaps we can use
it too, in that case we don't need to export do_wait().

```
> +void zap_pid_ns_processes(struct pid_namespace *pid_ns)
> +{
> +     int nr;
> +     int rc;
> +     int options = WEXITED|__WALL;
> +
> +     /*
```

```
> + * We know pid == 1 is terminating. Find remaining pid_ts
> + * in the namespace, signal them and then wait for them
> + * exit.
> + */
> + nr = next_pidmap(pid_ns, 1);
> + while (nr > 0) {
> +   kill_proc_info(SIGKILL, SEND_SIG_PRIV, nr);
> +   nr = next_pidmap(pid_ns, nr);
> + }
```

Without tasklist_lock held this is not reliable.

Oleg.
