

---

Subject: Re: [PATCH] Use struct pid parameter in copy\_process()  
Posted by [Sukadev Bhattiprolu](#) on Fri, 23 Feb 2007 23:26:57 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Eric W. Biederman [ebiederm@xmission.com] wrote:

| sukadev@us.ibm.com writes:

|  
| > From: Sukadev Bhattiprolu <sukadev@us.ibm.com>  
| > Subject: [PATCH] Use struct pid parameter in copy\_process()  
| >  
| > 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.

| I would recommend doing this in 2 steps:

- | - One patch to kill the likely(p->pid).
- | - And another to kill change the pid argument.

Yes. I can break that up into two patches, but I missed and Badari pointed the other caller to copy\_process()

```
struct task_struct * __cpuinit fork_idle(int cpu)
{
    struct task_struct *task;
    struct pt_regs regs;

    task = copy_process(CLONE_VM, 0, idle_regs(&regs), 0, NULL, NULL, 0);
    if (!IS_ERR(task))
        init_idle(task, cpu);

    return task;
}
```

Now this is passing a null struct pid which would not be good if I remove the if (likely(p->pid)) check in copy\_process().

Does this copy\_process() mean there can be multiple tasks with pid\_t == 0 (one per cpu on an SMP system) ?

Can we simply attach all those tasks to init\_struct\_pid by passing in &init\_struct\_pid to the above copy\_process() ?

|  
| The indentation change makes it really hard to see what  
| the change in pid argument buys.

Right.

|  
| This also needs to be part of the patchset that adds a dummy  
| struct pid to init, to make the dependency clear.

Ok.

|  
| Also given that you change the type there is no need to change  
| the name of the pid parameter to copy process, and the spid  
| name just looks strange.

Ok.

|  
| Eric

---

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

---