

---

Subject: Re: [PATCH] cgroup: fix a race condition in manipulating tsk->cg\_list

Posted by [Paul Menage](#) on Thu, 17 Apr 2008 04:17:34 GMT

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

---

On Wed, Apr 16, 2008 at 9:11 PM, Andrew Morton

<akpm@linux-foundation.org> wrote:

>

> I don't fully understand the race. Both paths hold css\_set\_lock.

>

> Can you describe it in more detail please?

Task A starts exiting, passes the check for unlinking current->cg\_list.

Before it completely exits task B does the very first cgroup\_iter\_begin() call (via reading a cgroups tasks file) which links all tasks in to their css\_set objects via tsk->cg\_list.

Then task A finishes exiting and is freed, but doesn't unlink from the cg\_list.

>

> afacit the task at \*p could set PF\_EXITING immediately after this code has

> tested PF\_EXITING and then the task at \*p could proceed until we hit the

> same race (whatever that is).

The important fact there is that the task sets PF\_EXITING \*before\* it checks whether it needs to unlink from current->cg\_list.

Paul

---

Containers mailing list

Containers@lists.linux-foundation.org

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

---