Subject: Re: [RFC][PATCH] Do not set /proc inode->pid for non-pid-related inodes
Posted by ebiederm on Thu, 22 Mar 2007 12:16:08 GMT
View Forum Message <> Reply to Message

Cedric Le Goater <clg@fr.ibm.com> writes:

> [long long thread]
>
> Eric W. Biederman wrote:
>> Cedric Le Goater <clg@fr.ibm.com> writes:
>>
>>>>> what about a kthread that would be spawned when a task is cloned in an
>>>>> unshared pid namespace ? This is an extra cost in term of tasks.
>>>> If you use kernel_thread this can happen. (Die kernel_thread).
>>>> If you use the kthread interface keventd will be the parent process and
>>>> we won't have problems.
>>> so is it something acceptable for mainline ? I think openvz has such
>>> a thread doing the reaping.
>>
>> Please clarify. Is what acceptable for mainline?
>
> [As i kind of jumped in the thread, i did some digging in the thread to
>   see where these comments were coming from.]
>
> Correct me if i got something wrong : the initial question is how do we
> handle the pid namespace exit and if we mandate task with pid == 1 to be
> the last task to die ?
>
> So I suggested to have a kthread be pid == 1 for each new pid namespace.
> the kthread can do the killing of all tasks if needed and will die when
> the refcount on the pid namespace == 0.
>
> Would such a (rough) design be acceptable for mainline ?

The case that preserves existing semantics requires us to be able to
run /sbin/init in a container. Therefore pid 1 should be a user space
process.

So I don't think a design that doesn't allow us to run /sbin/init as
in a container would be acceptable for mainline.

Eric

____
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers