Subject: Re: [RFC][PATCH] Add child reaper to struct pspace
Posted by serue on Fri, 08 Sep 2006 13:25:00 GMT
View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
> Sukadev Bhattiprolu <sukadev@us.ibm.com> writes:
>
> > Cedric Le Goater [clg@fr.ibm.com] wrote:
> > |
> > | <snip>
> > |
> > | > */
> > | > static void
> > | > forget_original_parent(struct task_struct *father, struct list_head
> > *to_release)
> > | > @@ -669,7 +670,7 @@ forget_original_parent(struct task_struc
> > | > do {
> > | > reaper = next_thread(reaper);
> > | > if (reaper == father) {
> > | > - reaper = child_reaper;
> > | > + reaper = father->pspace->child_reaper;
> > | > break;
> > | > }
> > | > } while (reaper->exit_state);
> > | > @@ -857,7 +858,7 @@ fastcall NORET_TYPE void do_exit(long co
> > |
> > | what about killing all the task in that pid space if child_reaper == init
> > | dies ?
> > |
> >
> > We probably need that for instance when a process in the parent pspace
> > kills the init of a child pspace, we should destroy the child pspace
> > by killing all the tasks in the child pspace including the child reaper.
> >
> > I guess we need to maintain a list of task_structs in the pspace and walk
> > that list. Will work on that as a separate patch.
>
> Yes. We all so need something like that list to support kill -1.
> Although walking the list of all processes may be sufficient for a first
> pass.
>
> The real trick is handing nested pid namespaces, properly.

Not if, as you've suggested in the past, pid_ns 5 has valid pids in its
own pid_ns for every process in pid_namespaces nested under it.

It should be simple to implement, should not impact the non-container
cases, and should only start to impact performance as the nesting gets

deep, which AFAIK we all believe won't happen (max nesting of 2 AFAICS,
one checkpointable application container under one vserver-thingie)

And it makes kill -1 trivial, as in pid_ns 5 we just kill all processes
in pid_ns 5, without worrying about finding the ones in it's decendent
pid namespaces.

-serge

_____