Subject: Re: Processes with multiple pid_t values
Posted by serue on Tue, 12 Dec 2006 16:04:55 GMT
View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
> Sukadev Bhattiprolu <sukadev@us.ibm.com> writes:
>
> > A process that unshares its namespace gets a new pid_t in the child
> > namespace. Simlarly its process group and session leaders get new pid_ts
> > in the child namespace right ?
> >
> > i.e do the following pid_ts look reasonable when process 1234 unshares
> > its pid namespace ?
> >
> >
> >   PID PPID PGID SID
> >
> > init pid ns 1234 1233 1230 1220
> >
> > child pid ns 3 2 1 0
>
> A slightly more complete answer.
>
> A pid that cannot be represented in the current pid namespace should be
> 0.
>
> pid 1 is very special and in the case of a clone should definitely
> be the first pid in the namespace.
>
> In the case of an unshare pid == 1 is probably the process that does
> the unshare, and it's children all show up in the child namespace.

Sigh, here we go back again to the question of what to do in the case
of a lightweight container which doesn't have a /sbin/init. Let's say
I do
 spawn_container(fork_and_exit(sleep 30m));

so pid 1 forks, pid 2 sleeps 30 minutes, but pid 1 exits right after
the fork. What do we do? Create an idle pid 1? Tack a struct pid
to the parent namespaces' pid=1 making it pid=1 for the child namespace?

-serge

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