Subject: Re: [PATCH 0/13] Pid namespaces (OpenVZ view)
Posted by serue on Thu, 24 May 2007 16:20:06 GMT
View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
> Pavel Emelianov <xemul@openvz.org> writes:
>
> > That's how OpenVZ sees the pid namespaces.
> >
> > The main idea is that kernel keeps operating with tasks pid
> > as it did before, but each task obtains one more pid for each
> > pid type - the virtual pid. When putting the pid to user or
> > getting the pid from it kernel operates with the virtual ones.
>
> Just a quick reaction.
>
> - I would very much like to see a minimum of 3 levels of pids,
>   being supported. Otherwise it is easy to overlook some of the
>   cases that are required to properly support nesting, which long
>   terms seems important.

Pavel,

If I wanted to start a virtual server and in there start some checkpoint
restart jobs, so I start a new pid namespace inside the c/r job, what
will happen?

 a. second pidns unshare is refused
 b. second pidns unshare is allowed, but c/r job is not visible
 from the virtual server (but is from the global pidns)
 c. second pidns unshare is allowed, and somehow the c/r job
 is visible from the virtual server

If (a), is this a short-term shortcoming for simplicity of prototype and
code review, or do you think it's actually the right thing t do long
term?

thanks,
-serge

> - Semantically fork is easier then unshare. Unshare can mean
>   a lot of things, and it is easy to pick a meaning that has weird
>   side effects. Your implementation has a serious problem in that you
>   change the value of getpid() at runtime. Glibc does not know how to
>   cope with the value of getpid() changing.
>
> Eric