Subject: Re: [PATCH 0/13] Pid namespaces (OpenVZ view)
Posted by xemul on Thu, 24 May 2007 16:11:30 GMT

View Forum Message <> Reply to Message

Serge E. Hallyn wrote:
> Quoting Pavel Emelianov (xemul@openvz.org):
>> 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.
>>
>> E.g. virtual pid is returned from getpid(), virtual pgid -
>> from getpgid() and so on. Getting virtual pid from user is
>> performed in setpgid(), setsid() and kill() mainly and in some
>> other places.
>>
>> As far as the namespace are concerned I propose the following
>> scheme. The namespace can be created from unshare syscall only.
>> This makes fork() code look easier. Of course task must be
>
> So is your main reason for posting this as a counter to Suka's patchset
> the concern of overhead at clone?

No, that's just a coincidence that I worked on the same problem.
What I propose is another way to make pid namespaces. It has its
advantages over Suka's approach. Main are:

1. Lighter exporting of pid to userspace and performance issues
   on the whole - as you have noticed at least fork() is
   lighter and many syscalls that return task pids are;
2. Kernel logic of tracking pids is kept - virtual pids are
   used on kernel-user boundary only;
3. Cleaner logic for namespace migration: with this approach
   one need to save the virtual pid and let global one change;
   with Suka's logic this is not clear how to migrate the level
   2 namespace (concerning init to be level 0).

> thanks,
> -serge