Subject: Re: Re: pspace child_reaper
Posted by ebiederm on Wed, 30 Aug 2006 13:42:43 GMT
View Forum Message <> Reply to Message

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

> Hello,
>
> Roman Kagan wrote:
>
> [...]
>
>>> As for the per-container init process, the alternative to always
>>> enforcing a separate init process for every container is to allow an
>>> option of making the process which did the pidspace unshare (or is it
>>> the parent of that process) masquerade as (pidspace=new_container, pid=1).
>>
>> There's no point enforcing a separate 'init' process in every container.
>> The root of the process tree in a namespace has to be the child reaper
>> for that namespace meaning that
>>
>> - it is immune to signals, ptracing, etc. from within the pidspace
>> - every process in the pidspace is reparented to it once that process'
>>   parent dies
>> - when it dies the whole pidspace is termiated
>
> That's how i feel also.

Those sound like the correct semantics. Although terminating all of
it's children in a given pid namespace is an interesting semantic to
implement. But it seems to be the only sane one. At least it
is better then the current version where the kernel exits if pid == 1
is terminated.

> The key point here is that the process becoming the init of that pidspace
> is immune to sigchlg : ignores them or garbage collects them or handles EINTR.
>
> If we feel confortable with the above, let's bring back this question to a
> user space issue : the process doing an unshare of this pidspace must
> handle the sigchld one way or the other.

Sounds good.

I'm not convinced an unshare of a pid namespace is a well defined
operation. But creating a new pid namespace at clone time certainly
is, and that is what replicates the python example.

Eric

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