Subject: Re: Processes with multiple pid_t values
Posted by ebiederm on Sat, 09 Dec 2006 08:13:51 GMT
View Forum Message <> Reply to Message

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 ?

So far the only place where switching to a new pid namespace has an
obvious answer is with sys clone. At that point the answer is:

PID == 1 PPID = 0 PGID = 1 SID = 1.

Just like /sbin/init gets.

>
> 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
>
> Assuming they are :-), we should expect find_pid() call with nr == 0, 1,
> or 2 in the child pid namespace to also work right ?

find_pid with nr == 0 will always fail.

> But processes 1220, 1230, 1233 are entered into the hash table based on
> their init pid ns values. And so the above find_pid() calls would not
> find the process we want.
>
> i.e some processes have two pid_ts and we want to find them using either
> of the two values. The pid_hash table can obviously hash on one value.
>
> We would need some serious changes to the pid_hash table to do this ???
> (can we change the hash algorithm to generate a key based on all pid_ts
> a process has ????)

You just enter a struct pid into the hash table based upon each pid_t value
assigned to it. You make the key comparison (namespace, pid_t) == (namespace, pid_t)
when doing the hash lookup.

> Or should we just keep track of these special processes (4 per namespace
> including the child reaper) in the namespace object - just like we treat
> the child_reaper special ?
>
> Then find_pid() would have to check the namespace object in addition to
> hash table.

Assume that we will have pid namespaces nested in pid namespaces 2 or 3 levels deep.

Understand what a session id, a process group id, and a pid are and how they are normally assigned please.

Don't even consider unshare of the pid namespace until you can see how to handle the simpler clone case.

Eric

_____

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