Subject: Re: Pid namespaces problems
Posted by ebiederm on Sun, 04 Nov 2007 04:06:48 GMT
View Forum Message <> Reply to Message

Pavel Emelyanov <xemul@openvz.org> writes:

> Hi, Eric, Suka.
>
> Eric, you and Ulrich claim that pid namespaces are full of BUGs.
> Can you please share you BUG list with me, so I could correct
> mine.

To be clear. I think the current pid namespace work is incomplete.
I do not think the pid namespaces is fundamentally buggy the way
Ingo and Ulrich were suggesting (my apologies for the delayed
reply I have been away from my computer).

I think I shared just about everything I know of off the top
of my head in earlier threads. But I haven't tried to
find an exhaustive list of uncorrected code as they pop
up fairly easily when I audit various pid users. Which
lead me to conclude that the pid namespace is not complete.

> Things as I see them now are the following:
> 1. signals delivery is not perfect in the namespace
> 2. fs/lock.c will report wrong ids in the namespace
> 3. some kernel threads (nfs) still use old api (relevant
> for a namespace only)
> 4. tsk->pid and tsk->tgid should not be explicitly used
A wrapper that gets those values for use in printk.

As a general principle I am opposed to using global pid values
(except in kernel print statements). Using them we continue
to have pid wrap around issues if we store them, and mixing
global pid_t values and non-global pid_t values is all too possible.

For example:
fs/autofs/inode.c: line 83 *pgrp = task_pgrp_nr(current);
fs/autofs/inode.c: line 117: *pgrp = option;

We are simultaneously assigning a global pid and a pid from
the current pid namespace to the same variable. Ouch!

Used in anything but the init_pid namespace that code is wrong.

So with stupid things like that I would very much like to
convert everything to storing and comparing struct pid pointers
which have essentially the same cost as pid_t values, can

be used the same way, but cannot be accidentally mixed with
with pid_t operations. So they are just less error prone.

> I'd appreciate some specific information, like "the ttys
> in drivers/char/tty_io.c may break the pid refcounting"
> rather than abstract "this is not clear whether the
> refcounting is good in fs/locks.c"

You are picking on the one instance that I figured I would need
further review to see if there was work that needed to be done. That
is what I meant by unclear. I didn't know if the code was safe and
the code wasn't using one of the idioms that would have made me
certain that the code was safe.

- We need to store a struct pid reference on the sysvipc semaphores (and
  probably the other sysvipc objects) so that if they are used across
  namespace boundaries we can convert and give processes the pid for
  their local namespace.

- There are several architectures with their own signal functions for
  other OS compatibility that have are using _pid and not _vpid
  variants of functions. (irix and solaris)
  arch/mips/kernel/irixsig.c:irix_waitsys
  arch/mips/kernel/sysirix.c:irix_setpgrp
  arch/sparc64/solaris/misc.c:solaris_procids

Eric

_____

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers