Subject: Re: Attaching PID 0 to a cgroup
Posted by Matt Helsley on Thu, 03 Jul 2008 21:59:35 GMT
View Forum Message <> Reply to Message

On Wed, 2008-07-02 at 03:24 +0530, Dhaval Giani wrote:
> On Tue, Jul 01, 2008 at 11:48:31PM +0200, Andrea Righi wrote:
> > Li Zefan wrote:
> >> CC: Paul Jackson <pj@sgi.com>
> >>
> >> Dhaval Giani wrote:
> >>> [put in the wrong alias for containers list correcting it.]
> >>>
> >>> On Tue, Jul 01, 2008 at 03:15:45PM +0530, Dhaval Giani wrote:
> >>>> Hi Paul,
> >>>>
> >>>> Attaching PID 0 to a cgroup caused the current task to be attached to
> >>>> the cgroup. Looking at the code,
> >>>>
> >>
> >> [...]
> >>
> >>>> I was wondering, why this was done. It seems to be unexpected behavior.
> >>>> Wouldn't something like the following be a better response? (I've used
> >>>> EINVAL, but I can change it to ESRCH if that is better.)
> >>>>
> >>
> >> Why is it unexpected? it follows the behavior of cpuset, so this patch will
> >> break backward compatibility of cpuset.
> >>
> >> But it's better to document this.
> >>
> >> ---
> >>
> >> Document the following cgroup usage:
> >> # echo 0 > /dev/cgroup/tasks
> >>
> >> Signed-off-by: Li Zefan <lizf@cn.fujitsu.com>
> >> ---
> >> cgroups.txt | 4 ++++
> >> 1 file changed, 4 insertions(+)
> >>
> >> diff --git a/Documentation/cgroups.txt b/Documentation/cgroups.txt
> >> index 824fc02..213f533 100644
> >> --- a/Documentation/cgroups.txt
> >> +++ b/Documentation/cgroups.txt
> >> @@ -390,6 +390,10 @@ If you have several tasks to attach, you have to do it one after another:
> >>   ...

> >> # /bin/echo PIDn > tasks
> >> +You can attach the current task by echoing 0:
> >> +
> >> +# /bin/echo 0 > tasks
> >> +
> >> 3. Kernel API
> >> ==========
> >
> > Wouldn't be more meaningful to specify the bash's builtin echo here
> > even if it doesn't opportunely handle write() errors?
> >
> > Using /bin/echo would attach /bin/echo itself to the cgroup, that just
> > exists, so it seems like a kind of noop, isn't it?
> >
>
> Yes, you are right. this example should use bash's builtin echo.

 IMHO you need to include this point in the docs verbosely rather than
just switching the docs to bash's builin-in echo. Otherwise it doesn't
fully resolve the fundamental confusion you correctly identified.

Or perhaps a snippet of simplified C code will make it clear:

```
------------
 char buffer[16];
 int fd;

 fd = open("/some/cgroup/tasks", O_WRONLY);

 /*
  * These two writes produce the same effect: adding this process
  * to /some/cgroup.
  */
 if (the_slightly_shorter_way)
  write(fd, "0", 2);
 else {
  /* The slightly-less-short way */
  snprintf(buffer, 16, "%u", getpid());
  write(fd, buffer, strlen(buffer));
 }
------------
```

Cheers,
 -Matt Helsley

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