Subject: Re: [RFC] Prefixing cgroup generic control filenames with "cgroup."
Posted by akpm on Thu, 28 Feb 2008 22:21:00 GMT
View Forum Message <> Reply to Message

On Thu, 28 Feb 2008 14:06:30 -0800
"Paul Menage" <menage@google.com> wrote:

```
> On Thu, Feb 28, 2008 at 1:40 PM, Andrew Morton
> <akpm@linux-foundation.org> wrote:
> >
> >  Maybe cgroups shouldn't be putting kernel-generated files in places where
> >  user-specified files appear?
> >
>
> Well, that API (mixing control files and group directories in the same
> directory namespace) was inherited directly from cpusets.
>
> It wouldn't be hard to throw that away and move all the user-created
> group directories into their own subdirectory, i.e. change the
> existing directory layout from something like:
>
> /mnt/cgroup/
>     tasks
>     cpu.shares
>     memory.limit_in_bytes
>     memory.usage_in_bytes
>     user_created_groupname1/
>         tasks
>         cpu.shares
>         memory.limit_in_bytes
>         memory.usage_in_bytes
>     user_created_groupname2/
>         tasks
>         cpu.shares
>         memory.limit_in_bytes
>         memory.usage_in_bytes
>
> to something like:
>
> /mnt/cgroup/
>     tasks
>     cpu.shares
>     memory.limit_in_bytes
>     memory.usage_in_bytes
>     groups/
>         user_created_groupname1/
>             tasks
>             cpu.shares
>           memory.limit_in_bytes
>           memory.usage_in_bytes
>           groups/
>        user_created_groupname2/
>           tasks
>           cpu.shares
>           memory.limit_in_bytes
>           memory.usage_in_bytes
>           groups/
```

That looks nice.

> That would completely solve the namespace problem, at the cost of a
> little extra verbosity/inelegance for human users (I suspect
> programmatic users would prefer it), and lack of compatibility with
> 2.6.24. I'd also need to make the existing model a mount option so
> that we could keep compatibility with the cpusets filesystem API.

That doesn't. It sounds like cpusets legacy has mucked us up here?

Could we do something like auto-prefixing user-created directories with a
fixed string so that there is no way in which the user can cause a
collision with kernel-created files?

I suppose that would break cpusets back-compatibility as well? If so, we
could do the prefixing only for non-cpusets directories, but that's getting
a bit weird.

> > (Am still thrashing around a bit here without an overview of the overall
> > layout and naming).
>
> Pretty much the same as cpusets, other than the additional
> kernel-generated files in each directory, as provided by the resource
> subsystems. So the same potential problem faced cpusets, but the fact
> that new cpuset features weren't being developed quickly meant the
> problem was less likely to actually bite people.

hm. I guess that all the kernel-generated file names are known up-front
and that they are instantiated early, so if a user tried to create a cgroup
called "tasks", than that would just fail.

But, as you say, later addition of new kernel-created files might collide
with prior userspace installations.

So yet another option would be to promise to prefix all _future_
kernel-generated files with "kern_", and to change the implementation now
to reject any user-created files which start with "kern_". hm.