Subject: Re: [RFC] [PATCH] Re: Prefixing cgroup generic control filenames with "cgroup."
Posted by Paul Jackson on Fri, 29 Feb 2008 15:30:13 GMT
View Forum Message <> Reply to Message

Paul M wrote:
> Well, the additional components are called "groups" not "cgroups", but
> basically yes.

Ah yes - "groups" - right, sorry:

/mnt/cgroup/groups/user_created_groupname1/groups/user_created_groupname2

> Yes, we could just say "the kernel reserves the right to use any names
> that begin with a lower-case letter, and no others", and be done with
> it.

The pattern might be stronger (more restrictive) than "[a-z].*"

The pattern might be something like:
1) the known set of existing names (a short, specific list), plus
2) "[a-z]+\.[a-z]+(_[a-z]+)*"

That second pattern is some lower case letters, a dot, and some
more lower case letters, possibly with some embedded underscores.

That is, except for the grandfathered existing known names, such as
"tasks", you would be promising that all names added in the future
would look like the examples (for some string of lower case letters
"subsys"):

subsys.foo
subsys.bar_baz

or for cgroup infrastructure names (using "kern" or "groups" prefix,
I don't have a clear preference):

groups.stuff
groups.this_and_that

Then for instance any name (not already in use) that had either (1) no
embedded dot '.', or (2) at least one character other than "[a-z_.]+",
or (3) other variants too numerous to list, would be safe for user
created group names.

Or, for a simpler and more restrictive regex pattern, don't allow
underscores, resulting in all kernel generated names matching:

1) the known list, or
2) "[a-z]+\.[a-z]+"

Note here "safe for user created" names just means safe from collision with kernel generated names, not necessarily safe from collision with other user generated names.

That is regardless of what you do here, you cannot protect the delicate user from possible collision. You can only protect them from collision with "your" names.

This risks imposing extra complexity on users just so you can avoid being blamed for the name collisions they might still experience anyway. When I phrase it that way, my enthusiasm for this proposal weakens further.

> > And did I say incompatible with released versions?
>
> Not at all incompatible if it requires a mount option to enable it ...

Ah - are you saying I missed another detail? That depending on how they mounted it, the path might be either of:

/mnt/cgroup/groups/user_created_groupname1/groups/user_created_groupname2
or
/mnt/cgroup/user_created_groupname1/user_created_groupname2

So code that knows something about these paths has to work with either form (since not all code using these paths will control the mount relevant option.)

I hope I misunderstood something here.

> That leaves a bit of an ugly taste in my mouth ...

Could you spell out the key reason -you- find it distasteful, perhaps for a stronger pattern such as "[a-z]+\.[a-z]+" I consider above? Perhaps I'm missing some reason to share in your revulsion.

> but if people seem to prefer that approach we can go for it.

So long as /dev/cpuset is unscathed, I'm ok either way. Let's see what others think.

--

I won't rest till it's the best ...
Programmer, Linux Scalability
Paul Jackson <pj@sgi.com> 1.940.382.4214

_____

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