Subject: Re: [PATCH 12/15] driver core: Implement tagged directory support for device classes.
Posted by ebiederm on Wed, 16 Jul 2008 21:09:55 GMT
View Forum Message <> Reply to Message

Tejun Heo <htejun@gmail.com> writes:
>> To do that I believe we would need to ensure sysfs does not use
>> the inode->i_mutex lock except to keep the VFS layer out. Allowing us
>> to safely change the directory structure, without holding it.
>
> I don't think sysfs is depending on i_mutex anymore but I need to go
> through the code to make sure.

The vfs still does. So at least for directory tree manipulation we
need to hold i_mutex before we grab sysfs_mutex.

I think that means we need to unscramble the whole set of locking
order issues.

In lookup we have:
local_vfs_lock -> fs_global_lock

In modifications we have:
fs_global_lock -> local_vfs_lock

Which is the definition of a lock ordering problem.

Currently we play jump through some significant hoops to keep things
in local_vfs_lock -> fs_global_lock order.

If we also take the rename_mutex on directory adds and deletes we
may be able to keep jumping through those hoops. However I expect
we would be in a much better situation if we could figure out how
to avoid the problem.

It looks like the easy way to handle this is to make the sysfs_dirent
list rcu protected. Which means we can fix our lock ordering problem
without VFS modifications. Allowing the locking to always
be: sysfs_mutex ... i_mutex.

After that it would be safe and a good idea to have unshared
inodes between superblocks, just so we don't surprise anyone
making generic VFS assumptions.

Eric

_____
Containers mailing list
Containers@lists.linux-foundation.org