Subject: Re: [PATCH 2/4] sysfs: Implement sysfs manged shadow directory support.
Posted by Tejun Heo on Tue, 31 Jul 2007 04:28:55 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> What do we use inode->i_mutex for? I think we might be able
> to kill that.
>
> I'm starting to wonder if we can completely remove sysfs
> from grabbing inode->i_mutex.

i_mutex is grabbed when dentry and inode locking requires it. It's not
used to protect sysfs internal data structure anymore. I don't think we
can remove i_mutex grabbing without violating dentry/inode locking rules.

>>> At first glance sysfs_assoc_lock looks just as bad.
>> I think sysfs_assoc_lock is okay. It's tricky tho. Why do you think
>> it's bad?
>
> I'm still looking. I just have a weird vibe so far. sysfs_get_dentry
> is really nasty with respect to locking.

Yes, sysfs_get_dentry() is pretty hairy. I wish I could use
path_lookup() there but can't allocate memory for path name because
looking up must succeed when it's called from removal path if dentry
already exists. Also, lookup_one_len_kern() bypasses security checks
and there's no equivalent path_lookup() like function which does that.

Locking rule aruond sysfs_assoc_lock is tricky. It's mainly used to
avoid race condition between sysfs_d_iput() vs. dentry creation, node
removal, etc. As long as sysfs_assoc_lock is held, sd->s_dentry can be
dereferenced but you also need dcache_lock to determine whether the
dentry is alive (dentry->d_inode != NULL) or in the process of being
killed. There were two or three race conditions around dentry
reclamation in the past and several discussion threads about them.

Thanks.

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