

Tejun Heo <htejun@gmail.com> writes:

```
>> +{
>> + struct sysfs_rename_struct *srs;
>> + struct super_block *sb;
>> + struct dentry *dentry;
>> + int error;
>> +
>> + list_for_each_entry(sb, &sysfs_fs_type.fs_supers, s_instances) {
>> +     dentry = sysfs_get_dentry(sb, sd);
>> +     if (!dentry)
>> +         continue;
>
> sysfs_get_dentry() return ERR_PTR() value. Oops, sysfs_get_dentry()
> implementation is wrong too. Also, please move
> sysfs_grab/release_supers() near this patch and add (a lot of)
> comments there.
>
> Other than that, I think this is as clean as this can be. Great.
```

Welcome. I will see what I can do with respect to cleaning up the names.

As for the return value of sysfs_get_dentry that is tricky. In particular I have three specific cases the code needs to deal with.

- We got the dentry.
- We did not get the dentry because for this super block there never ever will be a dentry.
- Some kind of error occurred in attempting to get the dentry.

Not getting a dentry because it is impossible I am currently handling with a NULL return. I can equally use a specific error code to mean that as well. It doesn't much matter. So I guess the hunk in question could read:

```
>> + list_for_each_entry(sb, &sysfs_fs_type.fs_supers, s_instances) {
>> +     dentry = sysfs_get_dentry(sb, sd);
>> +     if (dentry == ERR_PTR(-ENOENT))
>> +         continue;
```

As long as we handle that class of error differently I really don't care.

Eric

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>
