
Subject: [PATCH 13/14] sysfs: Rewrite rename in terms of sysfs dirents

Posted by [Tejun Heo](#) on Mon, 20 Aug 2007 12:36:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

From: Eric W. Biederman <ebiederm@xmission.com>

This patch rewrites sysfs_rename_dir to perform it's checks as much as possible on the underlying sysfs_dirents instead of the contents of the dcache. It turns out that this version is a little simpler, and a little more like the rest of the sysfs directory modification code.

tj: fixed double locking of sysfs_mutex

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

Signed-off-by: Tejun Heo <htejun@gmail.com>

fs/sysfs/dir.c | 38 ++++++-----

1 files changed, 16 insertions(+), 22 deletions(-)

diff --git a/fs/sysfs/dir.c b/fs/sysfs/dir.c

index 1752342..6432a94 100644

--- a/fs/sysfs/dir.c

+++ b/fs/sysfs/dir.c

@@ -767,7 +767,7 @@ void sysfs_remove_dir(struct kobject * kobj)

int sysfs_rename_dir(struct kobject * kobj, const char *new_name)
{

- struct sysfs_dirent *sd;

+ struct sysfs_dirent *sd = kobj->sd;

struct dentry *parent = NULL;

struct dentry *old_dentry = NULL, *new_dentry = NULL;

const char *dup_name = NULL;

@@ -775,63 +775,57 @@ int sysfs_rename_dir(struct kobject * kobj, const char *new_name)

mutex_lock(&sysfs_rename_mutex);

+ error = 0;

+ if (strcmp(sd->s_name, new_name) == 0)

+ goto out; /* nothing to rename */

+

/* get the original dentry */

- sd = kobj->sd;

old_dentry = sysfs_get_dentry(sd);

if (IS_ERR(old_dentry)) {

error = PTR_ERR(old_dentry);

- goto out_dput;

+ goto out;

```

}

parent = old_dentry->d_parent;

/* lock parent and get dentry for new name */
mutex_lock(&parent->d_inode->i_mutex);
+ mutex_lock(&sysfs_mutex);

- new_dentry = lookup_one_len(new_name, parent, strlen(new_name));
- if (IS_ERR(new_dentry)) {
-   error = PTR_ERR(new_dentry);
-   goto out_unlock;
- }
-
- error = -EINVAL;
- if (old_dentry == new_dentry)
+ error = -EEXIST;
+ if (sysfs_find_dirent(sd->s_parent, new_name))
    goto out_unlock;

- error = -EEXIST;
- if (new_dentry->d_inode)
+ error = -ENOMEM;
+ new_dentry = d_alloc_name(parent, new_name);
+ if (!new_dentry)
    goto out_unlock;

/* rename kobject and sysfs_dirent */
error = -ENOMEM;
new_name = dup_name = kstrdup(new_name, GFP_KERNEL);
if (!new_name)
-   goto out_drop;
+   goto out_unlock;

error = kobject_set_name(kobj, "%s", new_name);
if (error)
-   goto out_drop;
+   goto out_unlock;

- mutex_lock(&sysfs_mutex);
  dup_name = sd->s_name;
  sd->s_name = new_name;
- mutex_unlock(&sysfs_mutex);

/* rename */
d_add(new_dentry, NULL);
d_move(old_dentry, new_dentry);

```

```
error = 0;
- goto out_unlock;
-
- out_drop:
- d_drop(new_dentry);
out_unlock:
+ mutex_unlock(&sysfs_mutex);
mutex_unlock(&parent->d_inode->i_mutex);
- out_dput:
kfree(dup_name);
dput(old_dentry);
dput(new_dentry);
+ out:
mutex_unlock(&sysfs_rename_mutex);
return error;
}
--
1.5.0.3
```

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