
Subject: [PATCH 08/14] sysfs: Perform renames under sysfs_mutex

Posted by [ebiederm](#) on Tue, 31 Jul 2007 10:29:46 GMT

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

This modifies sysfs_rename_dir to use sysfs_addrm_start/sysfs_addrm_finish making it more like the rest of the sysfs directory manipulation functions.

This moves the entire rename operation inside of sysfs_mutex removing the need to grab i_mutex, and incidentally this kills one s_dentry use.

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

fs/sysfs/dir.c | 46 ++++++-----

1 files changed, 17 insertions(+), 29 deletions(-)

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

index f22d60c..8e614d3 100644

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

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

@@ -827,7 +827,7 @@ void sysfs_remove_dir(struct kobject * kobj)

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

```
{
    struct sysfs_dirent *sd = kobj->sd;
- struct dentry *new_parent = NULL;
+ struct sysfs_addrm_cxt acxt;
    struct dentry *old_dentry = NULL, *new_dentry = NULL;
    const char *dup_name = NULL;
    int error;
```

@@ -836,59 +836,47 @@ int sysfs_rename_dir(struct kobject *kobj, const char *new_name)

```
    old_dentry = sysfs_get_dentry(sd);
```

```
    if (IS_ERR(old_dentry)) {
        error = PTR_ERR(old_dentry);
```

```
- goto out_dput;
```

```
+ goto out;
```

```
    }
```

```
- new_parent = dget(old_dentry->d_parent);
```

```
-
```

```
- /* lock new_parent and get dentry for new name */
```

```
- mutex_lock(&new_parent->d_inode->i_mutex);
```

```
-
```

```
- new_dentry = lookup_one_len(new_name, new_parent, strlen(new_name));
```

```
- if (IS_ERR(new_dentry)) {
```

```
-     error = PTR_ERR(new_dentry);
```

```
-     goto out_unlock;
```

```
- }
```

```
+ sysfs_addrm_start(&acxt, sd->s_parent);
```

```

    error = -EINVAL;
- if (old_dentry == new_dentry)
- goto out_unlock;
+ if (strcmp(new_name, sd->s_name) == 0)
+ goto addrm_finish;

    error = -EEXIST;
- if (new_dentry->d_inode)
- goto out_unlock;
+ if (sysfs_find_dirent(acxt.parent_sd, new_name))
+ goto addrm_finish;
+
+ new_dentry = d_alloc_name(old_dentry->d_parent, new_name);
+ if (!new_dentry)
+ goto addrm_finish;

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

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

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

    /* move under the new parent */
    d_add(new_dentry, NULL);
- d_move(sd->s_dentry, new_dentry);
+ d_move(old_dentry, new_dentry);

    error = 0;
- goto out_unlock;
-
- out_drop:
- d_drop(new_dentry);
- out_unlock:
- mutex_unlock(&new_parent->d_inode->i_mutex);
- out_dput:
+ addrm_finish:
+ sysfs_addrm_finish(&acxt);

```

```
kfree(dup_name);  
- dput(new_parent);  
  dput(old_dentry);  
  dput(new_dentry);  
+out:  
  return error;  
}
```

```
--  
1.5.1.1.181.g2de0
```

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