
Subject: [PATCH 11/25] sysfs: Remove s_dentry
Posted by [ebiederm](#) on Tue, 07 Aug 2007 21:19:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

The only uses of s_dentry left are the code that maintains s_dentry and trivial users that don't actually need it. So this patch removes the s_dentry maintenance code and restructures the trivial uses to use something else.

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

```
fs/sysfs/dir.c | 32 ++++-----  
fs/sysfs/mount.c | 1 -  
fs/sysfs/sysfs.h | 1 -  
3 files changed, 4 insertions(+), 30 deletions(-)
```

```
diff --git a/fs/sysfs/dir.c b/fs/sysfs/dir.c  
index adb1b01..1078e60 100644
```

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

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

```
@@ -247,22 +247,7 @@ static void sysfs_d_iput(struct dentry * dentry, struct inode * inode)  
{  
    struct sysfs_dirent * sd = dentry->d_fsdata;
```

```
- if (sd) {  
- /* sd->s_dentry is protected with sysfs_assoc_lock.  
- * This allows sysfs_drop_dentry() to dereference it.  
- */  
- spin_lock(&sysfs_assoc_lock);  
-  
- /* The dentry might have been deleted or another  
- * lookup could have happened updating sd->s_dentry to  
- * point the new dentry. Ignore if it isn't pointing  
- * to this dentry.  
- */  
- if (sd->s_dentry == dentry)  
- sd->s_dentry = NULL;  
- spin_unlock(&sysfs_assoc_lock);  
- sysfs_put(sd);  
- }  
+ sysfs_put(sd);  
    iput(inode);  
}
```

```
@@ -310,9 +295,6 @@ struct sysfs_dirent *sysfs_new_dirent(const char *name, umode_t mode,  
int type)
```

```
    * @sd: target sysfs_dirent
```

```
    * @dentry: dentry to associate
```

```

*
- * Associate @sd with @dentry. This is protected by
- * sysfs_assoc_lock to avoid race with sysfs_d_iput().
- *
* LOCKING:
* mutex_lock(sysfs_mutex)
*/
@@ -320,12 +302,6 @@ static void sysfs_attach_dentry(struct sysfs_dirent *sd, struct dentry
*dentry)
{
    dentry->d_op = &sysfs_dentry_ops;
    dentry->d_fsdata = sysfs_get(sd);
-
- /* protect sd->s_dentry against sysfs_d_iput */
- spin_lock(&sysfs_assoc_lock);
- sd->s_dentry = dentry;
- spin_unlock(&sysfs_assoc_lock);
-
    d_rehash(dentry);
}

@@ -927,7 +903,7 @@ int sysfs_rename_dir(struct kobject * kobj, const char *new_name)

/* rename */
d_add(new_dentry, NULL);
- d_move(sd->s_dentry, new_dentry);
+ d_move(old_dentry, new_dentry);

error = 0;
goto out_unlock;
@@ -960,7 +936,7 @@ int sysfs_move_dir(struct kobject *kobj, struct kobject *new_parent_kobj)
    error = PTR_ERR(old_dentry);
    goto out_dput;
}
- old_parent = sd->s_parent->s_dentry;
+ old_parent = old_dentry->d_parent;

new_parent = sysfs_get_dentry(new_parent_sd);
if (IS_ERR(new_parent)) {
@@ -986,7 +962,7 @@ again:
} else
    error = 0;
d_add(new_dentry, NULL);
- d_move(sd->s_dentry, new_dentry);
+ d_move(old_dentry, new_dentry);
dput(new_dentry);

/* Remove from old parent's list and insert into new parent's list. */

```

```

diff --git a/fs/sysfs/mount.c b/fs/sysfs/mount.c
index 9fae7d5..4968d31 100644
--- a/fs/sysfs/mount.c
+++ b/fs/sysfs/mount.c
@@ -57,7 +57,6 @@ static int sysfs_fill_super(struct super_block *sb, void *data, int silent)
    iput(inode);
    return -ENOMEM;
}
- sysfs_root.s_dentry = root;
  root->d_fsdata = &sysfs_root;
  sb->s_root = root;
  return 0;
diff --git a/fs/sysfs/sysfs.h b/fs/sysfs/sysfs.h
index 77253aa..8ed13cf 100644
--- a/fs/sysfs/sysfs.h
+++ b/fs/sysfs/sysfs.h
@@ -37,7 +37,6 @@ struct sysfs_dirent {
    unsigned int s_flags;
    umode_t s_mode;
    ino_t s_ino;
- struct dentry * s_dentry;
    struct iattr * s_iattr;
    atomic_t s_event;
};
--
1.5.1.1.181.g2de0

```

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