

---

Subject: [PATCH 10/14] sysfs: Rework sysfs\_drop\_dentry  
Posted by [ebiederm](#) on Tue, 31 Jul 2007 10:37:34 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

This modifies sysfs\_drop\_dentry so it no longer needs the sysfs\_assoc\_lock and its assorted challenges.

The principal idea is that \_\_sysfs\_get\_dentry when not creating dentry will return a dentry for the sysfs\_dirent if there is one in the dcache. If we do get a dentry we can drop it and force it out of the cache.

For good measure I am also calling shrink\_dcache\_parent which will force all of the child dentries out of the dcache as well if it is a directory.

This is essentially an adaption of proc\_flush\_task from procfs to handle the similar problems of sysfs.

Signed-off-by: Eric W. Biederman <[ebiederm@xmission.com](mailto:ebiederm@xmission.com)>

---

fs/sysfs/dir.c | 34 ++++++++-----  
1 files changed, 12 insertions(+), 22 deletions(-)

diff --git a/fs/sysfs/dir.c b/fs/sysfs/dir.c  
index ed2e6f3..f8de6fb 100644

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

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

@@ -516,35 +516,25 @@ void sysfs\_remove\_one(struct sysfs\_addrm\_cxt \*acxt, struct sysfs\_dirent \*sd)

    \* parent on entry to this function such that it can't be looked  
    \* up anymore.  
    \*

- \* @sd->s\_dentry which is protected with sysfs\_assoc\_lock points  
- \* to the currently associated dentry but we're not holding a  
- \* reference to it and racing with dput(). Grab dcache\_lock and  
- \* verify dentry before dropping it. If @sd->s\_dentry is NULL or  
- \* dput() beats us, no need to bother.  
+ \* We find the dentry for @sd by probing the dcache while holding  
+ \* sysfs\_mutex to keep it from changing until we are done.  
    \*/

static void sysfs\_drop\_dentry(struct sysfs\_dirent \*sd)  
{  
    struct dentry \*dentry = NULL;  
    struct inode \*inode;

- /\* We're not holding a reference to ->s\_dentry dentry but the  
- \* field will stay valid as long as sysfs\_assoc\_lock is held.

```

- */
- spin_lock(&sysfs_assoc_lock);
- spin_lock(&dcache_lock);
-
- /* drop dentry if it's there and dput() didn't kill it yet */
- if (sd->s_dentry && sd->s_dentry->d_inode) {
-     dentry = dget_locked(sd->s_dentry);
-     spin_lock(&dentry->d_lock);
-     __d_drop(dentry);
-     spin_unlock(&dentry->d_lock);
+ /* Remove a dentry for a sd from the dcache if present */
+ mutex_lock(&sysfs_mutex);
+ dentry = __sysfs_get_dentry(sd, 0);
+ if (IS_ERR(dentry))
+     dentry = NULL;
+ if (dentry) {
+     shrink_dcache_parent(dentry);
+     d_drop(dentry);
+     dput(dentry);
+ }
-
- spin_unlock(&dcache_lock);
- spin_unlock(&sysfs_assoc_lock);
-
- dput(dentry);
+ mutex_unlock(&sysfs_mutex);

/* adjust nlink and update timestamp */
inode = ilookup(sysfs_sb, sd->s_ino);
--
1.5.1.1.181.g2de0

```

---

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

---