Subject: [PATCH 12/25] sysfs: Introduce sysfs_rename_mutex
Posted by ebiederm on Tue, 07 Aug 2007 21:21:06 GMT
View Forum Message <> Reply to Message

Looking carefully at the rename code we have a subtle dependency
that the structure of sysfs not change while we are performing
a rename. If the parent directory of the object we are renaming
changes while the rename is being performed nasty things could
happen when we go to release our locks.

So introduce a sysfs_rename_mutex to prevent this highly
unlikely theoretical issue.

In addition hold sysfs_rename_mutex over all calls to
sysfs_get_dentry. Allowing sysfs_get_dentry to be simplified
in the future.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 fs/sysfs/dir.c | 8 +++++++-
 fs/sysfs/file.c | 4 ++++
 fs/sysfs/sysfs.h | 1 +
 3 files changed, 12 insertions(+), 1 deletions(-)

diff --git a/fs/sysfs/dir.c b/fs/sysfs/dir.c
index 1078e60..e0f49b7 100644
--- a/fs/sysfs/dir.c
+++ b/fs/sysfs/dir.c
@@ -15,6 +15,7 @@
 #include "sysfs.h"

 DEFINE_MUTEX(sysfs_mutex);
+DEFINE_MUTEX(sysfs_rename_mutex);
 spinlock_t sysfs_assoc_lock = SPIN_LOCK_UNLOCKED;

 static spinlock_t sysfs_ino_lock = SPIN_LOCK_UNLOCKED;
@@ -774,7 +775,7 @@ static struct dentry *__sysfs_get_dentry(struct super_block *sb, struct
sysfs_di
  * down from there looking up dentry for each step.
  *
  * LOCKING:
- * Kernel thread context (may sleep)
+ * mutex_lock(sysfs_rename_mutex)
  *
  * RETURNS:
  * Pointer to found dentry on success, ERR_PTR() value on error.
@@ -859,6 +860,8 @@ int sysfs_rename_dir(struct kobject * kobj, const char *new_name)
  const char *dup_name = NULL;

```
  int error;

+ mutex_lock(&sysfs_rename_mutex);
+
  /* get the original dentry */
  sd = kobj->sd;
  old_dentry = sysfs_get_dentry(sd);
@@ -916,6 +919,7 @@ int sysfs_rename_dir(struct kobject * kobj, const char *new_name)
  kfree(dup_name);
  dput(old_dentry);
  dput(new_dentry);
+ mutex_unlock(&sysfs_rename_mutex);
  return error;
 }

@@ -927,6 +931,7 @@ int sysfs_move_dir(struct kobject *kobj, struct kobject *new_parent_kobj)
  struct dentry *old_dentry = NULL, *new_dentry = NULL;
  int error;

+ mutex_lock(&sysfs_rename_mutex);
  BUG_ON(!sd->s_parent);
  new_parent_sd = new_parent_kobj->sd ? new_parent_kobj->sd : &sysfs_root;

@@ -983,6 +988,7 @@ again:
  dput(new_parent);
  dput(old_dentry);
  dput(new_dentry);
+ mutex_unlock(&sysfs_rename_mutex);
  return error;
 }

diff --git a/fs/sysfs/file.c b/fs/sysfs/file.c
index 416351a..fe783ea 100644
--- a/fs/sysfs/file.c
+++ b/fs/sysfs/file.c
@@ -470,7 +470,9 @@ int sysfs_update_file(struct kobject * kobj, const struct attribute * attr)
  if (!victim_sd)
   goto out;

+ mutex_lock(&sysfs_rename_mutex);
  victim = sysfs_get_dentry(victim_sd);
+ mutex_unlock(&sysfs_rename_mutex);
  if (IS_ERR(victim)) {
   rc = PTR_ERR(victim);
   victim = NULL;
@@ -509,7 +511,9 @@ int sysfs_chmod_file(struct kobject *kobj, struct attribute *attr, mode_t
mode)
  if (!victim_sd)
```

```
   goto out;

+ mutex_lock(&sysfs_rename_mutex);
  victim = sysfs_get_dentry(victim_sd);
+ mutex_unlock(&sysfs_rename_mutex);
  if (IS_ERR(victim)) {
   rc = PTR_ERR(victim);
   victim = NULL;
diff --git a/fs/sysfs/sysfs.h b/fs/sysfs/sysfs.h
index 8ed13cf..791b3ed 100644
--- a/fs/sysfs/sysfs.h
+++ b/fs/sysfs/sysfs.h
@@ -89,6 +89,7 @@ extern int sysfs_setattr(struct dentry *dentry, struct iattr *iattr);

 extern spinlock_t sysfs_assoc_lock;
 extern struct mutex sysfs_mutex;
+extern struct mutex sysfs_rename_mutex;
 extern struct super_block * sysfs_sb;
 extern const struct file_operations sysfs_dir_operations;
 extern const struct file_operations sysfs_file_operations;
--
1.5.1.1.181.g2de0
```

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