

[patch] lockdep: annotate vfs_rmdir for filesystems that take i_mutex in delete_inode

[patch] lockdep: annotate vfs_rmdir for filesystems that take i_mutex in delete_inode

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-07/msg07248.html>

- *From:* Arjan van de Ven <arjan@xxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 26 Jul 2006 08:47:21 +0200
-

On Tue, 2006-07-25 at 22:33 -0700, Andrew Morton wrote:

On Wed, 26 Jul 2006 00:16:42 +0200

The VFS takes the directory i_mutex and reiserfs_delete_inode() takes the to-be-deleted file's i_mutex.

That's notabug and lockdep will need to be taught about it.

[2nd try, now with coffee]

This is another 3 level locking ordering:

do_rmdir takes the mutex of the parent directory

vfs_rmdir takes the mutex of the victim

shrink_dcache_parent ends up in the reiser delete_inode which takes the mutex of dead children of the victim

the I_MUTEX ordering rules are

I_MUTEX_PARENT -> I_MUTEX_CHILD -> <normal>

do_rmdir already has I_MUTEX_PARENT, delete_inode does <normal> so
vfs_rmdir needs I_MUTEX_CHILD (which is also logical)

Signed-off-by: Arjan van de Ven <arjan@xxxxxxxxxxxxxxxxx>

Index: linux-2.6.18-rc2-git5/fs/namei.c

--- linux-2.6.18-rc2-git5.orig/fs/namei.c

+++ linux-2.6.18-rc2-git5/fs/namei.c

@@ -1967,7 +1967,7 @@ int vfs_rmdir(struct inode *dir, struct

DQUOT_INIT(dir);

- mutex_lock(&dentry->d_inode->i_mutex);

+ mutex_lock_nested(&dentry->d_inode->i_mutex, I_MUTEX_CHILD);

dentry_unhash(dentry);

if (d_mountpoint(dentry))

[patch] lockdep: annotate vfs_rmdir for filesystems that take i_mutex in delete_inode

[patch] lockdep: annotate vfs_rmdir for filesystems that take i_mutex in delete_inode

error = -EBUSY;

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>