
Subject: [PATCH 07/25] elevate writer count for chown and friends

Posted by [Dave Hansen](#) on Mon, 11 Dec 2006 22:30:06 GMT

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

chown/chmod,etc... don't call permission in the same way that the normal "open for write" calls do. They still write to the filesystem, so bump the write count during these operations.

Signed-off-by: Dave Hansen <haveblue@us.ibm.com>

lxc-dave/fs/open.c | 37 ++++++-----

1 file changed, 33 insertions(+), 4 deletions(-)

diff -puN fs/open.c~06-24-elevate-writer-count-for-chown-and-friends fs/open.c

--- lxc/fs/open.c~06-24-elevate-writer-count-for-chown-and-friends 2006-12-11

14:22:01.000000000 -0800

+++ lxc-dave/fs/open.c 2006-12-11 14:22:01.000000000 -0800

@@ -511,9 +511,12 @@ asmlinkage long sys_fchmod(unsigned int

err = -EROFS;

if (IS_RDONLY(inode))

goto out_putf;

+ err = mnt_want_write(file->f_vfsmnt);

+ if (err)

+ goto out_putf;

err = -EPERM;

if (IS_IMMUTABLE(inode) || IS_APPEND(inode))

- goto out_putf;

+ goto out_drop_write;

mutex_lock(&inode->i_mutex);

if (mode == (mode_t) -1)

mode = inode->i_mode;

@@ -522,6 +525,8 @@ asmlinkage long sys_fchmod(unsigned int

err = notify_change(dentry, &newattrs);

mutex_unlock(&inode->i_mutex);

+out_drop_write:

+ mnt_drop_write(file->f_vfsmnt);

out_putf:

fput(file);

out:

@@ -541,13 +546,16 @@ asmlinkage long sys_fchmodat(int dfd, co

goto out;

inode = nd.dentry->d_inode;

+ error = mnt_want_write(nd.mnt);

+ if (error)

```

+ goto dput_and_out;
  error = -EROFS;
  if (IS_RDONLY(inode))
- goto dput_and_out;
+ goto out_drop_write;

  error = -EPERM;
  if (IS_IMMUTABLE(inode) || IS_APPEND(inode))
- goto dput_and_out;
+ goto out_drop_write;

  mutex_lock(&inode->i_mutex);
  if (mode == (mode_t) -1)
@@ -557,6 +565,8 @@ asmlinkage long sys_fchmodat(int dfd, co
  error = notify_change(nd.dentry, &newattrs);
  mutex_unlock(&inode->i_mutex);

+out_drop_write:
+ mnt_drop_write(nd.mnt);
dput_and_out:
  path_release(&nd);
out:
@@ -582,7 +592,7 @@ static int chown_common(struct dentry *
  error = -EROFS;
  if (IS_RDONLY(inode))
    goto out;
- error = -EPERM;
+ error = -EPERM;
  if (IS_IMMUTABLE(inode) || IS_APPEND(inode))
    goto out;
  newattrs.ia_valid = ATTR_CTIME;
@@ -611,7 +621,12 @@ asmlinkage long sys_chown(const char __u
  error = user_path_walk(filename, &nd);
  if (error)
    goto out;
+ error = mnt_want_write(nd.mnt);
+ if (error)
+ goto out_release;
  error = chown_common(nd.dentry, user, group);
+ mnt_drop_write(nd.mnt);
+out_release:
  path_release(&nd);
out:
  return error;
@@ -631,7 +646,12 @@ asmlinkage long sys_fchownat(int dfd, co
  error = __user_walk_fd(dfd, filename, follow, &nd);
  if (error)
    goto out;

```

```

+ error = mnt_want_write(nd.mnt);
+ if (error)
+ goto out_release;
  error = chown_common(nd.dentry, user, group);
+ mnt_drop_write(nd.mnt);
+out_release:
  path_release(&nd);
out:
  return error;
@@ -645,7 +665,11 @@ asmlinkage long sys_lchown(const char __
  error = user_path_walk_link(filename, &nd);
  if (error)
    goto out;
+ error = mnt_want_write(nd.mnt);
+ if (error)
+ goto out_release;
  error = chown_common(nd.dentry, user, group);
+out_release:
  path_release(&nd);
out:
  return error;
@@ -662,9 +686,14 @@ asmlinkage long sys_fchown(unsigned int
  if (!file)
    goto out;

+ error = mnt_want_write(file->f_vfsmnt);
+ if (error)
+ goto out_fput;
  dentry = file->f_path.dentry;
  audit_inode(NULL, dentry->d_inode);
  error = chown_common(dentry, user, group);
+ mnt_drop_write(file->f_vfsmnt);
+out_fput:
  fput(file);
out:
  return error;

```

—

Containers mailing list
Containers@lists.osdl.org
<https://lists.osdl.org/mailman/listinfo/containers>
