
Subject: [PATCH 09/25] elevate write count during entire ncp_ioctl()

Posted by [Dave Hansen](#) on Mon, 11 Dec 2006 22:30:08 GMT

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

Some ioctls need write access, but others don't. Make a helper function to decide when write access is needed, and take it.

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

lxc-dave/fs/ncpfs/ioctl.c | 55 ++++++
1 file changed, 54 insertions(+), 1 deletion(-)

diff -puN fs/ncpfs/ioctl.c~08-24-elevate-write-count-during-entire-ncp_ioctl fs/ncpfs/ioctl.c

--- lxc/fs/ncpfs/ioctl.c~08-24-elevate-write-count-during-entire-ncp_ioctl 2006-12-11

14:22:02.000000000 -0800

+++ lxc-dave/fs/ncpfs/ioctl.c 2006-12-11 14:22:02.000000000 -0800

@@ -14,6 +14,7 @@

#include <linux/ioctl.h>

#include <linux/time.h>

#include <linux/mm.h>

+#include <linux/mount.h>

#include <linux/highuid.h>

#include <linux/smp_lock.h>

#include <linux/vmalloc.h>

@@ -260,7 +261,7 @@ ncp_get_charsets(struct ncp_server* serv

}

#endif /* CONFIG_NCPFS_NLS */

-int ncp_ioctl(struct inode *inode, struct file *filp,

+static int __ncp_ioctl(struct inode *inode, struct file *filp,
 unsigned int cmd, unsigned long arg)

{

 struct ncp_server *server = NCP_SERVER(inode);

@@ -821,6 +822,58 @@ outrel:

 return -EINVAL;

}

+static int ncp_ioctl_need_write(unsigned int cmd)

+{

+ switch (cmd) {

+ case NCP_IOC_GET_FS_INFO:

+ case NCP_IOC_GET_FS_INFO_V2:

+ case NCP_IOC_NCPREQUEST:

+ case NCP_IOC_SETDENTRYTTL:

+ case NCP_IOC_SIGN_INIT:

+ case NCP_IOC_LOCKUNLOCK:

+ case NCP_IOC_SET_SIGN_WANTED:

```

+ return 1;
+ case NCP_IOC_GETOBJECTNAME:
+ case NCP_IOC_SETOBJECTNAME:
+ case NCP_IOC_GETPRIVATEDATA:
+ case NCP_IOC_SETPRIVATEDATA:
+ case NCP_IOC_SETCHARSETS:
+ case NCP_IOC_GETCHARSETS:
+ case NCP_IOC_CONN_LOGGED_IN:
+ case NCP_IOC_GETDENTRYTTL:
+ case NCP_IOC_GETMOUNTUID2:
+ case NCP_IOC_SIGN_WANTED:
+ case NCP_IOC_GETROOT:
+ case NCP_IOC_SETROOT:
+ return 0;
+ default:
+ /* unkown IOCTL command, assume write */
+ WARN_ON(1);
+ }
+ return 1;
+}
+
+int ncp_ioctl(struct inode *inode, struct file *filp,
+    unsigned int cmd, unsigned long arg)
+{
+ int ret;
+
+ if (ncp_ioctl_need_write(cmd)) {
+ /*
+  * inside the ioctl(), any failures which
+  * are because of file_permission() are
+  * -EACCESS, so it seems consistent to keep
+  * that here.
+  */
+ if (mnt_want_write(filp->f_vfsmnt))
+ return -EACCES;
+ }
+ ret = __ncp_ioctl(inode, filp, cmd, arg);
+ if (ncp_ioctl_need_write(cmd))
+ mnt_drop_write(filp->f_vfsmnt);
+ return ret;
+}
+
+#ifdef CONFIG_COMPAT
long ncp_compat_ioctl(struct file *file, unsigned int cmd, unsigned long arg)
{
—

```

Containers mailing list

