
Subject: [PATCH] Leases can be hidden by flocks

Posted by [Pavel Emelianov](#) on Mon, 10 Sep 2007 14:16:29 GMT

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

The inode->i_flock list contains the leases, flocks and posix locks in the specified order. However, the flocks are added in the head of this list thus hiding the leases from F_GETLEASE command, from time_out_leases() and other code that expects the leases to come first.

The following example will demonstrate this:

```
#define _GNU_SOURCE
```

```
#include <unistd.h>
```

```
#include <fcntl.h>
```

```
#include <stdio.h>
```

```
#include <sys/file.h>
```

```
static void show_lease(int fd)
```

```
{
```

```
    int res;
```

```
    res = fcntl(fd, F_GETLEASE);
```

```
    switch (res) {
```

```
        case F_RDLCK:
```

```
            printf("Read lease\n");
```

```
            break;
```

```
        case F_WRLCK:
```

```
            printf("Write lease\n");
```

```
            break;
```

```
        case F_UNLCK:
```

```
            printf("No leases\n");
```

```
            break;
```

```
        default:
```

```
            printf("Some shit\n");
```

```
            break;
```

```
    }
```

```
}
```

```
int main(int argc, char **argv)
```

```
{
```

```
    int fd, res;
```

```
    fd = open(argv[1], O_RDONLY);
```

```
    if (fd == -1) {
```

```
        perror("Can't open file");
```

```
        return 1;
```

```

    }

    res = fcntl(fd, F_SETLEASE, F_WRLCK);
    if (res == -1) {
        perror("Can't set lease");
        return 1;
    }

    show_lease(fd);

    if (flock(fd, LOCK_SH) == -1) {
        perror("Can't flock shared");
        return 1;
    }

    show_lease(fd);

    return 0;
}

```

The first call to `show_lease()` will show the write lease set, but the second will show no leases.

Fix the flock adding so that the leases always stay in the head of this list.

Found during making the flocks pid-namespaces aware.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

```

diff --git a/fs/locks.c b/fs/locks.c
index 6068f82..0db1a14 100644
--- a/fs/locks.c
+++ b/fs/locks.c
@@ -781,7 +781,7 @@ find_conflict:
     if (request->fl_flags & FL_ACCESS)
         goto out;
     locks_copy_lock(new_fl, request);
-    locks_insert_lock(&inode->i_flock, new_fl);
+    locks_insert_lock(before, new_fl);
     new_fl = NULL;
     error = 0;

```
