
Subject: [PATCH 2/3] lutimesat: extend do_utimes() with flags

Posted by [adobriyan](#) on Fri, 26 Jan 2007 11:16:14 GMT

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

Passing AT_SYMLINK_NOFOLLOW will stop following symlinks.

Pass 0 to get usual behaviour.

Signed-off-by: Alexey Dobriyan <adobriyan@openvz.org>

```
arch/alpha/kernel/osf_sys.c      | 2 +-
arch/sparc64/kernel/sys_sparc32.c | 2 +-
fs/compat.c                      | 4 ++-
fs/utimes.c                      | 15 ++++++++-----
include/linux/time.h             | 2 +-
5 files changed, 15 insertions(+), 10 deletions(-)
```

--- a/arch/alpha/kernel/osf_sys.c

+++ b/arch/alpha/kernel/osf_sys.c

```
@@ -963,7 +963,7 @@ osf_utimes(char __user *filename, struct
    return -EFAULT;
}
```

```
- return do_utimes(AT_FDCWD, filename, tvs ? ktvs : NULL);
```

```
+ return do_utimes(AT_FDCWD, filename, tvs ? ktvs : NULL, 0);
}
```

```
#define MAX_SELECT_SECONDS \
```

--- a/arch/sparc64/kernel/sys_sparc32.c

+++ b/arch/sparc64/kernel/sys_sparc32.c

```
@@ -847,7 +847,7 @@ asmlinkage long sys32_utimes(char __user
    return -EFAULT;
}
```

```
- return do_utimes(AT_FDCWD, filename, (tvs ? &ktvs[0] : NULL));
```

```
+ return do_utimes(AT_FDCWD, filename, (tvs ? &ktvs[0] : NULL), 0);
}
```

```
/* These are here just in case some old sparc32 binary calls it. */
```

--- a/fs/compat.c

+++ b/fs/compat.c

```
@@ -87,7 +87,7 @@ asmlinkage long compat_sys_utime(char __
    tv[0].tv_usec = 0;
    tv[1].tv_usec = 0;
}
```

```
- return do_utimes(AT_FDCWD, filename, t ? tv : NULL);
```

```
+ return do_utimes(AT_FDCWD, filename, t ? tv : NULL, 0);
}
```

```

asmlinkage long compat_sys_futimesat(unsigned int dfd, char __user *filename, struct
compat_timeval __user *t)
@@ -101,7 +101,7 @@ asmlinkage long compat_sys_futimesat(uns
    get_user(tv[1].tv_usec, &t[1].tv_usec))
    return -EFAULT;
}
- return do_utimes(dfd, filename, t ? tv : NULL);
+ return do_utimes(dfd, filename, t ? tv : NULL, 0);
}

asmlinkage long compat_sys_utimes(char __user *filename, struct compat_timeval __user *t)
--- a/fs/utimes.c
+++ b/fs/utimes.c
@@ -31,7 +31,7 @@ asmlinkage long sys_utime(char __user *
    tv[0].tv_usec = 0;
    tv[1].tv_usec = 0;
}
- return do_utimes(AT_FDCWD, filename, times ? tv : NULL);
+ return do_utimes(AT_FDCWD, filename, times ? tv : NULL, 0);
}

#endif
@@ -40,14 +40,19 @@ #endif
* must be owner or have write permission.
* Else, update from *times, must be owner or super user.
*/
-long do_utimes(int dfd, char __user *filename, struct timeval *times)
+long do_utimes(int dfd, char __user *filename, struct timeval *times, int flags)
{
- int error;
+ int error = -EINVAL;
    struct nameidata nd;
    struct inode * inode;
    struct iattr newattrs;
+ int follow;

- error = __user_walk_fd(dfd, filename, LOOKUP_FOLLOW, &nd);
+ if ((flags & ~AT_SYMLINK_NOFOLLOW) != 0)
+ goto out;
+
+ follow = (flags & AT_SYMLINK_NOFOLLOW) ? 0 : LOOKUP_FOLLOW;
+ error = __user_walk_fd(dfd, filename, follow, &nd);

    if (error)
        goto out;
@@ -93,7 +98,7 @@ asmlinkage long sys_futimesat(int dfd, c

```

```

    if (utimes && copy_from_user(&times, utimes, sizeof(times)))
        return -EFAULT;
- return do_utimes(dfd, filename, utimes ? times : NULL);
+ return do_utimes(dfd, filename, utimes ? times : NULL, 0);
}

asmlinkage long sys_utimes(char __user *filename, struct timeval __user *utimes)
--- a/include/linux/time.h
+++ b/include/linux/time.h
@@ -108,7 +108,7 @@ extern void do_gettimeofday(struct timev
extern int do_settimeofday(struct timespec *tv);
extern int do_sys_settimeofday(struct timespec *tv, struct timezone *tz);
#define do_posix_clock_monotonic_gettime(ts) ktime_get_ts(ts)
-extern long do_utimes(int dfd, char __user *filename, struct timeval *times);
+extern long do_utimes(int dfd, char __user *filename, struct timeval *times, int flags);
struct itimerval;
extern int do_setitimer(int which, struct itimerval *value,
    struct itimerval *ovalue);

```
