

---

Subject: [PATCH 4/3] lutimesat: compat syscall and wire up on x86\_64

Posted by [adobriyan](#) on Mon, 26 Feb 2007 16:43:08 GMT

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

---

[Apply after:

lutimesat-simplify-utime2.patch

lutimesat-extend-do\_utimes-with-flags.patch

lutimesat-actual-syscall-and-wire-up-on-i386.patch

Tweak compat syscall number to match normal syscall number if needed.]

Signed-off-by: Alexey Dobriyan <[adobriyan@openvz.org](mailto:adobriyan@openvz.org)>

---

```
arch/x86_64/ia32/ia32entry.S | 1 +
fs/compat.c                   | 14 ++++++++
fs/utimes.c                   | 2 +-
include/linux/time.h          | 1 +
4 files changed, 17 insertions(+), 1 deletion(-)
```

--- a/arch/x86\_64/ia32/ia32entry.S

+++ b/arch/x86\_64/ia32/ia32entry.S

@@ -719,4 +719,5 @@ #endif

.quad compat\_sys\_move\_pages

.quad sys\_getcpu

.quad sys\_epoll\_pwait

+ .quad compat\_sys\_lutimesat /\* 320 \*/

ia32\_syscall\_end:

--- a/fs/compat.c

+++ b/fs/compat.c

```
@@ -109,6 +109,20 @@ asmlinkage long compat_sys_utimes(char _
    return compat_sys_futimesat(AT_FDCWD, filename, t);
}
```

```
+asmlinkage long compat_sys_lutimesat(int dfd, char __user *filename, struct compat_timespec
__user *utimes)
```

```
+{
```

```
+ struct timespec ts[2];
```

```
+
```

```
+ if (utimes) {
```

```
+ if (get_user(ts[0].tv_sec, &utimes[0].tv_sec) ||
```

```
+ get_user(ts[0].tv_nsec, &utimes[0].tv_nsec) ||
```

```
+ get_user(ts[1].tv_sec, &utimes[1].tv_sec) ||
```

```
+ get_user(ts[1].tv_nsec, &utimes[1].tv_nsec))
```

```
+ return -EFAULT;
```

```
+ }
```

```
+ return do_utimes_nsec(dfd, filename, utimes ? ts : NULL, AT_SYMLINK_NOFOLLOW);
```

```
+}
```

```

+
asmlinkage long compat_sys_newstat(char __user * filename,
    struct compat_stat __user *statbuf)
{
--- a/fs/utimes.c
+++ b/fs/utimes.c
@@ -40,7 +40,7 @@ #endif
    * must be owner or have write permission.
    * Else, update from *times, must be owner or super user.
    */
-static long do_utimes_nsec(int dfd, char __user *filename, struct timespec *times, int flags)
+long do_utimes_nsec(int dfd, char __user *filename, struct timespec *times, int flags)
{
    int error = -EINVAL;
    struct nameidata nd;
--- a/include/linux/time.h
+++ b/include/linux/time.h
@@ -110,6 +110,7 @@ extern int do_settimeofday(struct timespec
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, int flags);
+extern long do_utimes_nsec(int dfd, char __user *filename, struct timespec *times, int flags);
struct itimerval;
extern int do_setitimer(int which, struct itimerval *value,
    struct itimerval *ovalue);

```

---

Subject: Re: [PATCH 4/3] lutimesat: compat syscall and wire up on x86\_64

Posted by [Stephen Rothwell](#) on Tue, 27 Feb 2007 03:06:24 GMT

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

---

On Mon, 26 Feb 2007 19:50:24 +0300 Alexey Dobriyan <adobriyan@openvz.org> wrote:

```

>
> +asmlinkage long compat_sys_lutimesat(int dfd, char __user *filename, struct compat_timespec
__user *utimes)
> +{
> + struct timespec ts[2];
> +
> + if (utimes) {
> + if (get_user(ts[0].tv_sec, &utimes[0].tv_sec) ||
> + get_user(ts[0].tv_nsec, &utimes[0].tv_nsec) ||
> + get_user(ts[1].tv_sec, &utimes[1].tv_sec) ||
> + get_user(ts[1].tv_nsec, &utimes[1].tv_nsec))
> + return -EFAULT;

if (get_compat_timespec(&ts[0], &utimes[0]) ||
    get_compat_timespec(&ts[1], &utimes[1]))
return -EFAULT;

```

--

Cheers,  
Stephen Rothwell                      sfr@canb.auug.org.au  
<http://www.canb.auug.org.au/~sfr/>

---

---

Subject: Re: [PATCH 4/3] lutimesat: compat syscall and wire up on x86\_64  
Posted by [adobriyan](#) on Tue, 27 Feb 2007 11:49:16 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Tue, Feb 27, 2007 at 02:06:24PM +1100, Stephen Rothwell wrote:  
> On Mon, 26 Feb 2007 19:50:24 +0300 Alexey Dobriyan <[adobriyan@openvz.org](mailto:adobriyan@openvz.org)> wrote:  
> >  
> > +asmlinkage long compat\_sys\_lutimesat(int dfd, char \_\_user \*filename, struct  
compat\_timespec \_\_user \*utimes)  
> > +{  
> > + struct timespec ts[2];  
> > +  
> > + if (utimes) {  
> > + if (get\_user(ts[0].tv\_sec, &utimes[0].tv\_sec) ||  
> > + get\_user(ts[0].tv\_nsec, &utimes[0].tv\_nsec) ||  
> > + get\_user(ts[1].tv\_sec, &utimes[1].tv\_sec) ||  
> > + get\_user(ts[1].tv\_nsec, &utimes[1].tv\_nsec))  
> > + return -EFAULT;  
>  
> if (get\_compat\_timespec(&ts[0], &utimes[0]) ||  
> get\_compat\_timespec(&ts[1], &utimes[1]))  
> return -EFAULT;

We have it? Thank you!

[PATCH 4/3] lutimesat: compat syscall and wire up on x86\_64

[Apply after:  
lutimesat-simplify-utime2.patch  
lutimesat-extend-do\_utimes-with-flags.patch  
lutimesat-actual-syscall-and-wire-up-on-i386.patch

Tweak compat syscall number to match normal syscall number if needed.]

Signed-off-by: Alexey Dobriyan <[adobriyan@openvz.org](mailto:adobriyan@openvz.org)>

---

```
arch/x86_64/ia32/ia32entry.S | 1 +
fs/compat.c                  | 12 ++++++++
fs/utimes.c                   | 2 +-
include/linux/time.h          | 1 +
```

4 files changed, 15 insertions(+), 1 deletion(-)

```
--- a/arch/x86_64/ia32/ia32entry.S
+++ b/arch/x86_64/ia32/ia32entry.S
@@ -719,4 +719,5 @@ #endif
 .quad compat_sys_move_pages
 .quad sys_getcpu
 .quad sys_epoll_pwait
+ .quad compat_sys_lutimesat /* 320 */
ia32_syscall_end:
--- a/fs/compat.c
+++ b/fs/compat.c
@@ -109,6 +109,18 @@ asmlinkage long compat_sys_utimes(char _
    return compat_sys_futimesat(AT_FDCWD, filename, t);
}

+asmlinkage long compat_sys_lutimesat(int dfd, char __user *filename, struct compat_timespec
+__user *utimes)
+{
+ struct timespec ts[2];
+
+ if (utimes) {
+ if (get_compat_timespec(&ts[0], &utimes[0]) ||
+ get_compat_timespec(&ts[1], &utimes[1]))
+ return -EFAULT;
+ }
+ return do_utimes_nsec(dfd, filename, utimes ? ts : NULL, AT_SYMLINK_NOFOLLOW);
+}
+
asmlinkage long compat_sys_newstat(char __user * filename,
    struct compat_stat __user *statbuf)
{
--- a/fs/utimes.c
+++ b/fs/utimes.c
@@ -40,7 +40,7 @@ #endif
 * must be owner or have write permission.
 * Else, update from *times, must be owner or super user.
 */
-static long do_utimes_nsec(int dfd, char __user *filename, struct timespec *times, int flags)
+long do_utimes_nsec(int dfd, char __user *filename, struct timespec *times, int flags)
{
    int error = -EINVAL;
    struct nameidata nd;
--- a/include/linux/time.h
+++ b/include/linux/time.h
@@ -110,6 +110,7 @@ extern int do_settimeofday(struct timesp
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, int flags);  
+extern long do_utimes_nsec(int dfd, char __user *filename, struct timespec *times, int flags);  
struct itimerval;  
extern int do_setitimer(int which, struct itimerval *value,  
    struct itimerval *ovalue);
```

---