Subject: [PATCH 1/3] lutimesat: simplify utime(2)
Posted by adobriyan on Fri, 26 Jan 2007 11:14:53 GMT
View Forum Message <> Reply to Message

Rewrite via do_utimes() like compat_sys_utime().

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

 fs/utimes.c | 50 +++++++--
 1 file changed, 7 insertions(+), 43 deletions(-)

--- a/fs/utimes.c
+++ b/fs/utimes.c
@@ -22,52 +22,16 @@ #ifdef __ARCH_WANT_SYS_UTIME
  */
 asmlinkage long sys_utime(char __user * filename, struct utimbuf __user * times)
 {
- int error;
- struct nameidata nd;
- struct inode * inode;
- struct iattr newattrs;
+ struct timeval tv[2];

- error = user_path_walk(filename, &nd);
- if (error)
- goto out;
- inode = nd.dentry->d_inode;
-
- error = -EROFS;
- if (IS_RDONLY(inode))
- goto dput_and_out;
-
- /* Don't worry, the checks are done in inode_change_ok() */
- newattrs.ia_valid = ATTR_CTIME | ATTR_MTIME | ATTR_ATIME;
  if (times) {
- error = -EPERM;
- if (IS_APPEND(inode) || IS_IMMUTABLE(inode))
-   goto dput_and_out;
-
- error = get_user(newattrs.ia_atime.tv_sec, &times->actime);
- newattrs.ia_atime.tv_nsec = 0;
- if (!error)
-   error = get_user(newattrs.ia_mtime.tv_sec, &times->modtime);
- newattrs.ia_mtime.tv_nsec = 0;
- if (error)
-   goto dput_and_out;
-

```
-  newattrs.ia_valid |= ATTR_ATIME_SET | ATTR_MTIME_SET;
- } else {
-               error = -EACCES;
-               if (IS_IMMUTABLE(inode))
-                       goto dput_and_out;
-
-  if (current->fsuid != inode->i_uid &&
-      (error = vfs_permission(&nd, MAY_WRITE)) != 0)
-   goto dput_and_out;
+  if (get_user(tv[0].tv_sec, &times->actime) ||
+      get_user(tv[1].tv_sec, &times->modtime))
+   return -EFAULT;
+  tv[0].tv_usec = 0;
+  tv[1].tv_usec = 0;
   }
- mutex_lock(&inode->i_mutex);
- error = notify_change(nd.dentry, &newattrs);
- mutex_unlock(&inode->i_mutex);
-dput_and_out:
- path_release(&nd);
-out:
- return error;
+ return do_utimes(AT_FDCWD, filename, times ? tv : NULL);
 }

 #endif
```

Subject: Re: [PATCH 1/3] lutimesat: simplify utime(2)
Posted by Andrew Morton on Fri, 26 Jan 2007 20:41:20 GMT
View Forum Message <> Reply to Message

On Fri, 26 Jan 2007 14:21:42 +0300
Alexey Dobriyan <adobriyan@openvz.org> wrote:

> Rewrite via do_utimes() like compat_sys_utime().
>
> Signed-off-by: Alexey Dobriyan <adobriyan@openvz.org>
> ---
>
> fs/utimes.c | 50 +++++++++--
> 1 file changed, 7 insertions(+), 43 deletions(-)
>
> --- a/fs/utimes.c
> +++ b/fs/utimes.c
> @@ -22,52 +22,16 @@ #ifdef __ARCH_WANT_SYS_UTIME
>   */
> asmlinkage long sys_utime(char __user * filename, struct utimbuf __user * times)

```
>  {
> - int error;
> - struct nameidata nd;
> - struct inode * inode;
> - struct iattr newattrs;
> + struct timeval tv[2];
>
> - error = user_path_walk(filename, &nd);
> - if (error)
> -  goto out;
> - inode = nd.dentry->d_inode;
> -
> - error = -EROFS;
> - if (IS_RDONLY(inode))
> -  goto dput_and_out;
> -
> - /* Don't worry, the checks are done in inode_change_ok() */
> - newattrs.ia_valid = ATTR_CTIME | ATTR_MTIME | ATTR_ATIME;
>   if (times) {
> -  error = -EPERM;
> -  if (IS_APPEND(inode) || IS_IMMUTABLE(inode))
> -   goto dput_and_out;
> -
> -  error = get_user(newattrs.ia_atime.tv_sec, &times->actime);
> -  newattrs.ia_atime.tv_nsec = 0;
> -  if (!error)
> -   error = get_user(newattrs.ia_mtime.tv_sec, &times->modtime);
> -  newattrs.ia_mtime.tv_nsec = 0;
> -  if (error)
> -   goto dput_and_out;
> -
> -  newattrs.ia_valid |= ATTR_ATIME_SET | ATTR_MTIME_SET;
> - } else {
> -               error = -EACCES;
> -               if (IS_IMMUTABLE(inode))
> -                       goto dput_and_out;
> -
> -  if (current->fsuid != inode->i_uid &&
> -      (error = vfs_permission(&nd, MAY_WRITE)) != 0)
> -   goto dput_and_out;
> +  if (get_user(tv[0].tv_sec, &times->actime) ||
> +      get_user(tv[1].tv_sec, &times->modtime))
> +   return -EFAULT;
> +  tv[0].tv_usec = 0;
> +  tv[1].tv_usec = 0;
>   }
> - mutex_lock(&inode->i_mutex);
> - error = notify_change(nd.dentry, &newattrs);
```

> - mutex_unlock(&inode->i_mutex);
> -dput_and_out:
> - path_release(&nd);
> -out:
> - return error;
> + return do_utimes(AT_FDCWD, filename, times ? tv : NULL);
> }
>
> #endif

I'm somewhat surprised that this wasn't done earlier. I wonder if there's
some subtle reason why this won't work. How well tested is this?

Subject: Re: [PATCH 1/3] lutimesat: simplify utime(2)
Posted by Arnd Bergmann on Fri, 26 Jan 2007 23:35:42 GMT
View Forum Message <> Reply to Message

On Friday 26 January 2007 21:41, Andrew Morton wrote:

http://www.opengroup.org/onlinepubs/000095399/functions/utim es.html
lists a slight difference between utime and utimes in the handling
of EPERM/EACCESS:

> The utimes() function shall fail if:
> [EACCES] Search permission is denied by a component of the path prefix;
> or the times argument is a null pointer and the effective user ID of the
> process does not match the owner of the file and write access is denied.
> [EPERM] The times argument is not a null pointer and the calling process'
> effective user ID has write access to the file but does not match the
> owner of the file and the calling process does not have the appropriate
> privileges.
>
> The utime() function shall fail if:
> [EACCES] Search permission is denied by a component of the path prefix;
> or the times argument is a null pointer and the effective user ID of the
> process does not match the owner of the file, the process does not have
> write permission for the file, and the process does not have appropriate
> privileges.
> [EPERM] The times argument is not a null pointer and the calling process'
> effective user ID does not match the owner of the file and the calling
> process does not have the appropriate privileges.

I don't really understand how that should be implemented in different
ways, but it might be the reason that we have separate functions.

Arnd <><

Subject: Re: [PATCH 1/3] lutimesat: simplify utime(2)
Posted by Alexey Dobriyan on Sun, 28 Jan 2007 15:24:59 GMT
View Forum Message <> Reply to Message

On Fri, Jan 26, 2007 at 12:41:20PM -0800, Andrew Morton wrote:
> On Fri, 26 Jan 2007 14:21:42 +0300
> Alexey Dobriyan <adobriyan@openvz.org> wrote:
>
> > Rewrite via do_utimes() like compat_sys_utime().
>
> I'm somewhat surprised that this wasn't done earlier.

Because, the following patch didn't hit -mm. :)

From adobriyan@gmail.com Wed Oct 25 20:32:24 2006
Date: Wed, 25 Oct 2006 20:32:24 +0400
From: Alexey Dobriyan <adobriyan@gmail.com>
To: Andrew Morton <akpm@osdl.org>
Cc: linux-kernel@vger.kernel.org
Subject: [PATCH] Shrink sys_utime()
Message-ID: <20061025163224.GA5356@martell.zuzino.mipt.ru>
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Disposition: inline
User-Agent: Mutt/1.5.11
Status: RO
Content-Length: 3314
Lines: 118

All checks in sys_utime() and do_utimes() are duplicated as well as a
comment. sys_utime() will now use do_utimes() after getting times from
userspace and projecting them to struct timeval [2].

Nevermind.

> I wonder if there's some subtle reason why this won't work.

I don't know. Compat syscall -- I'm not touching it. Normal syscall --
time_t is long on all archs, suseconds_t is sometimes int, but we're
putting zero there.

> How well tested is this?

It passed utime tests in December's LTP.

Subject: Re: [PATCH 1/3] lutimesat: simplify utime(2)
Posted by Alexey Dobriyan on Sun, 28 Jan 2007 15:28:42 GMT
View Forum Message <> Reply to Message

On Sat, Jan 27, 2007 at 12:35:42AM +0100, Arnd Bergmann wrote:
> On Friday 26 January 2007 21:41, Andrew Morton wrote:

>
> http://www.opengroup.org/onlinepubs/000095399/functions/utim es.html
> lists a slight difference between utime and utimes in the handling
> of EPERM/EACCESS:
>
> > The utimes() function shall fail if:
> > [EACCES] Search permission is denied by a component of the path prefix;
> > or the times argument is a null pointer and the effective user ID of the
> > process does not match the owner of the file and write access is denied.
> > [EPERM] The times argument is not a null pointer and the calling process'
> > effective user ID has write access to the file but does not match the
> > owner of the file and the calling process does not have the appropriate
> > privileges.
> >
> > The utime() function shall fail if:
> > [EACCES] Search permission is denied by a component of the path prefix;
> > or the times argument is a null pointer and the effective user ID of the
> > process does not match the owner of the file, the process does not have
> > write permission for the file, and the process does not have appropriate
> > privileges.
> > [EPERM] The times argument is not a null pointer and the calling process'
> > effective user ID does not match the owner of the file and the calling
> > process does not have the appropriate privileges.
>
> I don't really understand how that should be implemented in different
> ways, but it might be the reason that we have separate functions.

Present sys_utime() and do_utimes() are identical, except the former
does direct getusering into new attributes, and the latter accept "int
dfd" instead of hardcoded current working directory.