

## Re: what's next for the linux kernel?

**Source:** <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-10/1260.html>

---

**From:** Bodo Eggert (7eggert\_at\_gmx.de)

**Date:** 10/05/05

Date: Wed, 5 Oct 2005 23:05:34 +0200 (CEST)

To: Marc Perkel <marc@perkel.com>

On Wed, 5 Oct 2005, Marc Perkel wrote:

> *What you don't get is that if you don't have rights to write to a file  
> then you shouldn't have the right to delete the file.*

In unix, nobody but the kernel has the right to *\*delete\** a file. Therefore nobody can delete a file without write permission.

Files are deleted if the last reference is gone. If you play a music file and unlink it while it's playing, it won't be deleted until the player closes the file, since an open filehandle is a reference.

If you like, you can think of it as a kind of instant garbage collection:

Files are the objects referenced by these lists, and if you own the object, you can change it. However, as long as there is a reference, you can't destroy it, since this would invalidate all references. Instead, you must remove all references.

Directories are lists of references, and these lists are independent from the referenced file-objects. If you own the list, you can change it by adding or removing files. You can even link files not owned or accessible by you:

```
7eggert@be1:~/tmp > ls -l /tmp/foo/foo
----- 1 7eggert_b users 0 2005-10-05 22:32 /tmp/foo/foo
7eggert@be1:~/tmp > ln /tmp/foo/foo .
7eggert@be1:~/tmp > ls -l
total 0
----- 2 7eggert_b users 0 2005-10-05 22:32 foo
<snip>
```

Do you notice the link count in the second column?

Let's remove a link:

<snip>

Re: what's next for the linux kernel?

## Linux-Kernel: Re: what's next for the linux kernel?

```
*switch*
7eggert_b@be1:/tmp/foo> rm foo
rm: remove write-protected regular empty file `foo'? y
*switch*
7eggert@be1:~/tmp > ls -l
total 0
----- 1 7eggert_b users 0 2005-10-05 22:32 foo
<snip>
```

As you can see, each directory-owner can independantly unlink the file.

BTW: The owner can change the permissions on the linked file anytime, so even if you couldn't link non-accessable files, you could end up with entries in your private directory you could neither access nor delete.

I can also open the file as 7eggert\_b, delete it from another tty and still access it's contents:

```
<snip>
7eggert_b@be1:/tmp/foo> cat > foo
as
*switch*
be1:/home/7eggert/tmp # rm /tmp/foo/foo
be1:/home/7eggert/tmp # ll /proc/4820/fd/1
l-wx----- 1 7eggert_b users 64 Oct 5 22:43 /proc/4820/fd/1
-> /tmp/foo/foo (deleted)
be1:/home/7eggert/tmp # cat /proc/4820/fd/1
as
*switch*
df
be1:/home/7eggert/tmp # cat /proc/4820/fd/1
as
df
<snip>
```

As you can see, the directory entry is deleted, but the file is still there. However, making a hard link from a /proc/pid/fd entry is not (yet?) possible: "ln: creating hard link `baz' to `/proc/4927/fd/1': Invalid cross-device link".

```
--
Fun things to slip into your budget
Not in a budget, but in an annual report:
An employee stole 500,000+. They accounted for it on the annual report as
'involuntary employee relations expense'
-
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@vger.kernel.org
More majordomo info at http://vger.kernel.org/majordomo-info.html
Please read the FAQ at http://www.tux.org/lkml/
```