Filesystem and File Permissions*

Boris Veytsman

May 24, 2001

*This document contains lecture notes for informal Unix seminar for ITT AES employees (Reston, VA). No information in this document is either endorsed by or attributable to ITT. This document contains no ITT Privileged/Proprietary Information.

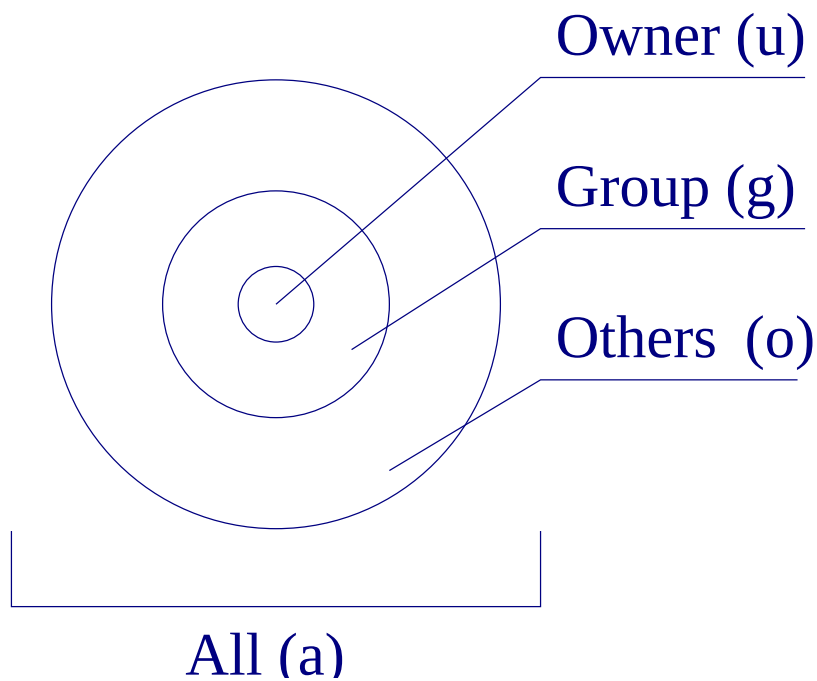
ITT Industries
Engineered for life

File Permissions in a Nutshell

For those of you in the reseller business, here is a helpful tip that will save your support staff a few hours of precious time. Before you send your next machine out to an untrained client, change the permissions on /etc/passwd to 666 and make sure there is a copy somewhere on the disk. Now when they forget the root password, you can easily login as an ordinary user and correct the damage. *CommUNIXque 1:1, ASCAR Business Systems*

ITT Industries
Engineered for life

Three rings for file permissions[1]:



[1] The root is above the rings

Three rights for the users:

Right 1: To read

Right 2: To write

Right 3: To execute

Example listing (from `ls-l`):

$$-\underbrace{\text{rwx}}_{\text{owner}}\underbrace{\text{r-x}}_{\text{group}}\underbrace{\text{--x}}_{\text{others}} \quad \underbrace{\text{boris}}_{\text{owner name}} \quad \underbrace{\text{staff}}_{\text{group name}} \quad \underbrace{\text{flist.pl}}_{\text{file name}}$$

Common Use

- One group per project
- Project directories are group-writable
- Everybody is a member of several groups

chmod, chown, chgrp

Not all who own a harp are harpers.
Marcus Terentius Varro

ITT Industries
Engineered for life

```
chown boris.staff flist.pl
```

- On some systems only root can do `chown`

```
chgrp staff flist.pl
```

```
chmod o-r flist.pl
chmod g+rw flist.pl
chmod u-w flist.pl
chmod -R g-r mail
```

Numeric Permissions

Your lucky number has been disconnected

Number	Action
4	Read
2	Write
1	Execute

Examples:

- $7 = 4 + 2 + 1$
- $6 = 4 + 2$
- $5 = 4 + 1$

ITT Industries
Engineered for life

Common permissions:

755: System Programs

644: System Libraries

664: Project Files

660: Secure Project files

644: User Files

600: Private User Files (mail, etc)

```
chmod 755 /var/Manhattan_Project/bin/*
```

Permissions of New Files: `umask`

> It's much easier to apologize than to get permission. *Grace Murray Hopper*

`umask` sets permissions you *do not* grant.

umask 000: Permission 777. Anybody can do anything

umask 022: Permission 755. For your non-private files

umask 002: Permission 775. For non-secure group projects.

GNU umask understands symbolic umask.

ITT Industries
Engineered for life

Group of the new file:

BSD classic: Same as the directory

SysV classic: Set by `newgrp`

Modern Systems: BSD if sgid is not set, SysV if is. Setting: `chmod g-s directory` or `chmod g+s directory`

Filesystems II: Advanced Information

Any sufficiently advanced technology is indistinguishable from magic. *Arthur C. Clarke*

- What does execution permission mean for a directory?
- Mike cannot write my files. How happens he can delete them?
- What are soft and hard links?
- I deleted a number of files, but no space was reclaimed. What happened?

Inodes and Metainformation

What is in a file?

Datablocks: The data
Inode: The *metainformation:* who owns the file, where data blocks are on a disk etc.
Directory: The file name

Directory is just a file with list of files and inodes. Files are not *inside* a directory, they are listed there!

```
boris@reston-0491:~/itt/unix/pdfs$ ls -a ./
.
..
3_files.pdf

boris@reston-0491:~/itt/unix/pdfs$ ls -ia ./
787745 .
442664 ..
787753 3_files.pdf
```

List of inodes & file names. Two special names: `.` and `..`

pdfs/

787745	.
442664	..
787753	3_files.pdf

Inode 787753

Owner: boris
Group: users
Permissions: 644
Modification time: ...
.....
Number of links: 1
Data blocks:

Filesystem and File Permissions*

Boris Veytsman

May 17, 2001

*This document contains lecture notes for informal Unix seminar for ITT AES employees (Reston, VA). No information in this document is either endorsed by or attributable to ITT. This document contains no ITT Privileged/Proprietary Information.

ITT Industries
Engineered for life

Read and Execute Bits for Directories

Read permission: you can read the *list* of files (`ls`)

Execute permission: you can read *inode* of a file (`ls -l`)

You do not know whether a file is readable/writable *unless* you have execute permission for the directory!

File Operations and Permissions

Operation	What Happens	Permissions	
		File	Directory
Reading	Reading inode & data	r	x
Writing	Data & inode change	w	x
Listing files	Reading directory data	-	r
"Long" listing	Reading directory & inode	-	rx
Executing program	Reading inode, getting data to kernel	x	x
Executing script	Reading inode & data	rx	x
Renaming	Directory change	-	w(x)
Deleting	Directory change & decrease of the number of links	-	w(x)

ITT Industries
Engineered for life

Question: A co-worker left a non-writable file in group projects directory. Can you change it? How?

Sticky Bit

`chmod 1777 tmp` or `chmod +t tmp`. Listing:

```
drwxrwxrwt boris staff tmp
```

Now you cannot delete files unless they are writable.

Operation	What Happens	Permissions	
		File	Directory
Renaming	Directory change	w	wx
Deleting	Directory change & decrease of the number of links	w	wx

Your mileage may vary!

ITT Industries
Engineered for life

Links: Hard and Soft

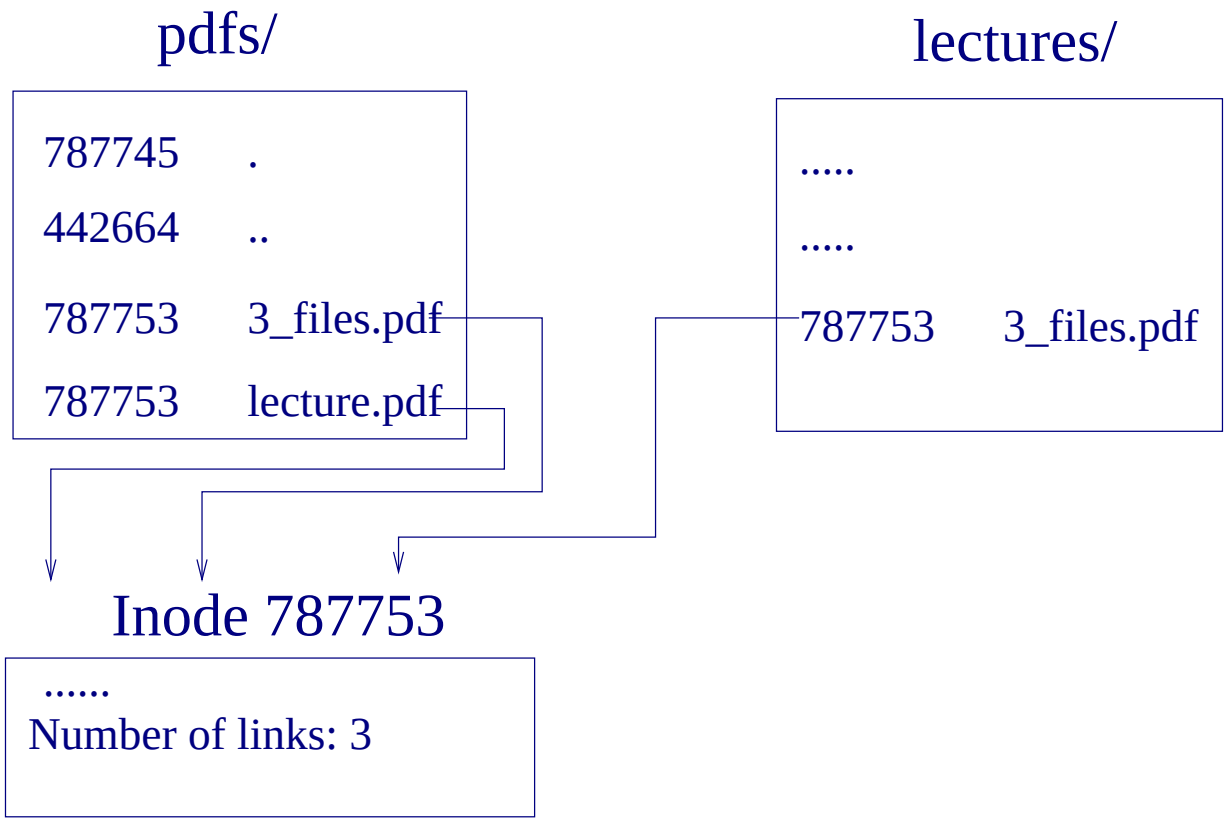
The perversity of nature is nowhere better demonstrated than by the fact that, when exposed to the same atmosphere, bread becomes hard while crackers become soft. *Clovis' Consideration of an Atmospheric Anomaly*

Why Links?

- Different programs might request different file names (`.xinit` and `.xsession`, or `libz.so`, `libz.so.1` and `libz.so.1.1.3`)
- Different programs might request different locations of libs
- You can write same program under different names:

```
if [ `basename $0` == "rmlink" ]
  then remove_link()
  else create_link()
fi
```

Hard Links

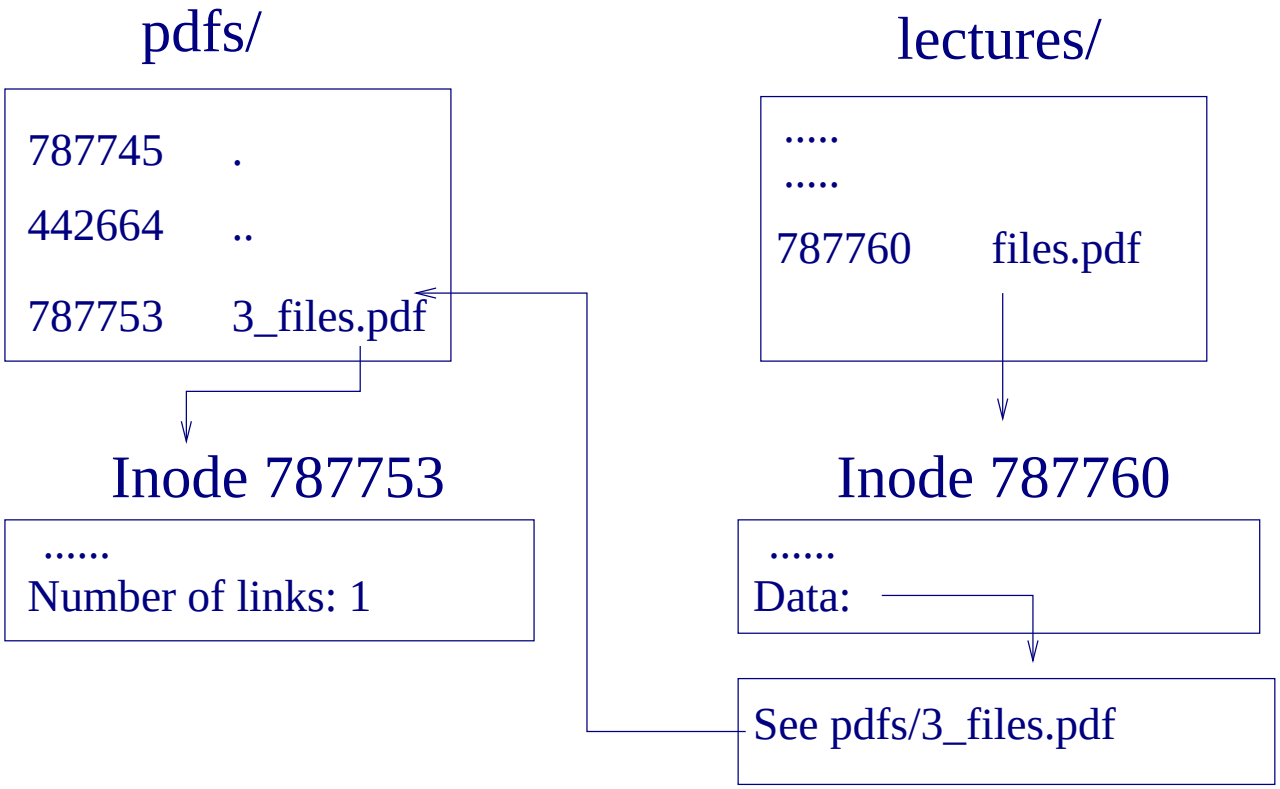

pdfs/
787745 .
442664 ..
787753 3_files.pdf
787753 lecture.pdf
lectures/
.....
.....
787753 3_files.pdf
Inode 787753
.....
Number of links: 3

- All hard links are *equal:* they are different names for the same file. They have the same permission (why?)
- Deleting a hard-linked file *does not* free space (why?)
- You cannot hard-link across file systems (why?)
- You cannot hard-link directories (why)
- Moving or renaming a hard link does not affect other hard links (why)

Creation:

```
ln source target
```

Symbolic Links



ITT Industries
Engineered for life

- Symbolic links are *not* equal to the original file; they are just aliases. They permissions are irrelevant.
- Deleting a soft link frees a very small amount of space
- You can soft-link across file systems and soft-link directories
- Moving or renaming a file makes all soft links *stale*

Creation:

```
ln -s source target
```

Question:

John had world-readable file `secrets/TopSecret.doc` in the world-readable directory `secrets`. After security lesson he said

```
chmod go-x secrets
```

He knew that to read a file you need an executable right for the directory and thought he is safe now. Nevertheless, he found out that Bob (found out to be a spy) *can* read the file `secrets/TopSecret.doc`. How did it happen?

Set `uid` and Set `gid` Privileges

Stinginess with privileges is kindness in disguise. *Guide to VAX/VMS Security, Sep. 1984*

Question: John wrote a program. Bob starts it. Whose files can it access?

Answer: Normally Bob's *unless* `suid` or `sgid` bites are set.

A program with `suid` or `sgid` is executed "on behalf of the owner"!

ITT Industries
Engineered for life

Why `suid` and `sgid`?

- You want to boost someone's privileges. Example: give everybody the right to shutdown
- You want to access system files

Security Concerns

- You give somebody a lot of rights!
- Badly written program can be coaxed to do unexpected things (buffer overflows)

The number of `suid` and `sgid` programs must be absolutely minimal!

If a program becomes unexpectedly `suid` or `sgid`—a hostile intrusion might be present!

Note 1: Due to race conditions scripts *cannot* be `suid` or `sgid`. Exception: suidperl.

Note 2: `sgid` for directories is a completely different matter!

How to Make a Program `suid` or `sgid`

```
chmod g+s myprogram
chmod u+s myprogram
chmod 6755 myprogram
```

Number	Action
4	Set `uid`
2	Set `gid`
1	Sticky bit

Listing:

```
-rwst-sr-x boris users flist.pl
```

ITT Industries
Engineered for life