

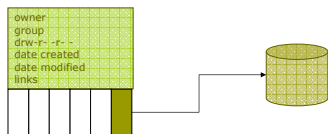
# CS-344 - Unix Operating System Fundamentals

## Lecture 6 Setting File and Directory Permissions

Based on slides created by Dr. Bangalore for the Spring 2005 offering of the course

### Unix File System: inodes

- Like an index card
- Contains all the information available about each file: owner, permissions, date created, etc.
- It also contains the location of the actual file on disk (data blocks on the hard drive)
- inodes numbers are four bytes long
- inodes are unique within a file system
- Use `ls -li` to see a list of files with their inodes



9/29/2006

3

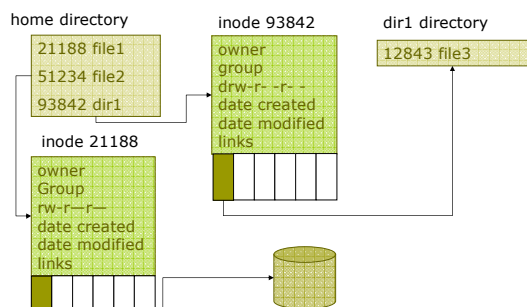
### Directories

- They are files with a list of file names and their corresponding inode numbers
- Special directories: `.` & `..`
  - `.` refers to itself
  - `..` refers to the parent directory
  - `/` is a special case
- Special environment variables (BASH)
  - `$PWD`
  - `$OLDPWD`
  - `cd - (linux)`
  - `pushd /directory and popd`

9/29/2006

4

### copy, move, and delete files



9/29/2006

5

### Working with links

	owner	group	date of modification	
<code>\$ ls -l</code>				
total 10				total # of data blocks
<code>-rw-r--r--</code>	1 puri	staff	2093 Feb 28 10:52 myfile	
<code>drwxr-xr-x</code>	3 puri	staff	4096 Feb 23 12:02 cs344	

- A listing for a file in a directory is a link to a file
- A file can be listed in more than one directory
- Two types of links:
  - Hard links
  - Symbolic links

9/29/2006

6

## Command Links

- *ln [options] existing-file new-file*
- Commonly used options:
  - -f → force creation of link, don't prompt if new file already exists
  - -n → don't create the link if new-file already exists
  - -s → creates a symbolic link to existing-file and named it new-file
- *ln* can be used with multiple files at the same time
- To remove a link
  - *rm link-name*

9/29/2006

7

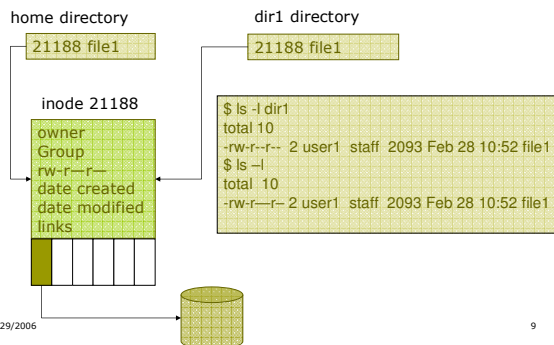
## Hard Links (I)

- The inode is hard-coded into the various directories
- Hard links can't be established between different file systems
- The # of links is incremented in all the directories
- The file is actually removed when it is removed from the last directory that lists it
- Only files can be linked in this way
- Only the superuser can create hard links for directories (and it doesn't work in all cases)

9/29/2006

8

## Hard Links (II)



9/29/2006

9

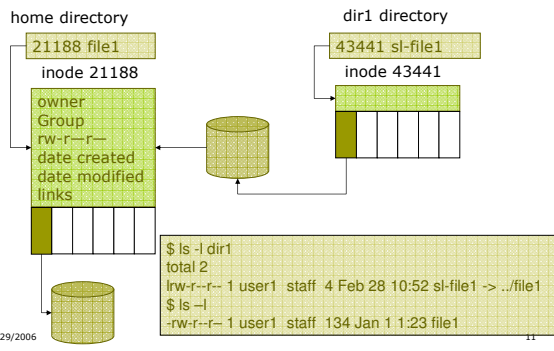
## Symbolic Links (I)

- Small file in the current directory that contains the information needed to locate the linked file wherever it is actually listed
- The original file and the symbolic links have their own inode
- Directories can be linked using symbolic links.
- If the original file or directory is removed or moved, the symbolic link points to nothing and it's broken

9/29/2006

10

## Symbolic Links (II)



9/29/2006

11

## File Permissions

- Typical UNIX user performs the following operations on files:
  - Read files (using *more*, *cat*, etc.)
  - Write files (using *>*, *>>*, *cat*, *vi*, etc.)
  - Execute commands in a file (shell scripts, executables, etc.)
- Correspondingly each file has three permissions read, write, and execute (*rwX*)
- On UNIX systems there are three classes of users: the *owner*, other members of *owner's group*, and all *other* users
- The owner can modify permissions for each of these three classes of users
- To examine file permissions use *ls -l*

```
$ ls -l myfile
-rw-r--r-- 1 puri staff 2093 Feb 28 10:52 myfile
$ ls -ld cs344
drwxr-xr-x 3 puri staff 4096 Feb 23 12:02 cs344
```

9/29/2006

12

## Determine user and group

- ❑ To determine login name of user type `"echo $USER"` or `"who am i"`
- ❑ To determine what groups you belong type `"groups"` (first group is your default group)
- ❑ To change to a new group type `"newgrp groupname"`, any new files created now will have this group name
- ❑ To determine your user id (UID) and group id (GID) type `"id"`
- ❑ To change group use the command `"chgrp"` and to change owner use the command `"chown"`

9/29/2006

13

## Changing File Permissions (I)

- ❑ To change current file permissions use `"chmod"` (change mode) command
- ❑ To add specific permission use `chmod +`
  - To add write permission to all users use:  
`chmod a+w filename`
  - To add read permission to only to users in your group use: `chmod g+r filename`
- ❑ To remove specific permission use `chmod -`
  - To remove read permission for all users use:  
`chmod a-r filename`
  - To remove read, write, and execute permission for the group and others use: `chmod go-rwx filename`
- ❑ You can also combine add and remove permissions (e.g., `chmod u+x,g+r,o-rwx filename`)

9/29/2006

14

## Changing File Permissions (II)

- ❑ Summary:

Types of Users		Permissions		Actions	
u	user	r	read	=	assign
g	group	w	write	-	remove
o	other	x	execute	+	add
a	all	-	none		

9/29/2006

15

## Using numerical permissions

- ❑ Instead of using u,g,o for user, group, and others we can also specify file permissions using numbers:
  - `rwX = 111 = 7`
  - `rw- = 110 = 6`
  - `r-x = 101 = 5`
  - `r-- = 100 = 4`
  - `-wx = 011 = 3`
  - `-w- = 010 = 2`
  - `--x = 001 = 1`
  - `--- = 000 = 0`
- ❑ `chmod go+rx filename = chmod 755 filename` (assuming current user permission is `rwX = 7`, if it is `rw- = 6`, then use `chmod 655 filename`)

9/29/2006

16

## Directory Permissions

- ❑ To list contents of a directory with `ls` command we need read permissions
- ❑ To add/remove files in a directory we need write and execute permissions
- ❑ To change to a directory or go through the directory we need execute permissions
- ❑ To list files with `ls -l` we need read and execute permissions for the directory, since information about permissions, owner, group, etc. are in the directory entry

9/29/2006

17

## Set/Get Default Directory Permissions

- ❑ When new file/directory is created the shell uses default permissions determined by `umask` value
- ❑ To obtain you default `umask` value, at command prompt enter: `umask`
- ❑ To change current `umask` value, enter `umask <new-mask-value>`
- ❑ Based on the `umask` value appropriate permissions are unmasked (allowed)
- ❑ Changing `umask` value has no effect on existing files, only new files will be effected

Umask	New Directory Permissions
000	rwXrwxrwx 777
022	rwXr-xr-x 755
027	rwXr-x-- 750
017	rwXrw---- 760

9/29/2006

18

## Get/Set File Permissions

- ❑ Set umask to *000*, create a new file, list the file using *ls -l*, this will indicate the default file permission (typically *rw-rw-rw-* = 666)
- ❑ Execute permission are never granted when files are created hence setting the mask on execute bit has no effect
- ❑ Set umask to *022*, create a new file, list file, the new file permissions will be *rw-r--r--* = 644
- ❑ Set umask to *023*, create a new file, list file, the new file permission will be still *rw-r--r--*
- ❑ To retain file permissions during file copy use *cp -p* option

9/29/2006

19

## File Permissions

```

$ ls -ld .
drwx--x--x 17 puri  faculty  8192 Feb 20 17:24 .
$ ls -l .bashrc
-rwx----- 1 puri  staff    1196 Feb 13 19:31 .bashrc
$ ls -ld /tmp/
drwxrwxrwt 17 root   sys     3218 Feb 20 18:35 /tmp/
$ umask
022
$ touch myfile
$ ls -l myfile
-rw-r--r-- 1 puri  staff    0 Feb 20 18:38 myfile
$ chmod +x myfile
$ ls -l myfile
-rwxr-xr-x 1 puri  staff    0 Feb 20 18:38 myfile
$ umask 027
$ touch newfile
$ ls -l newfile
-rw-r----- 1 puri  staff    0 Feb 20 18:39 newfile
$

```

9/29/2006

20