

The File System

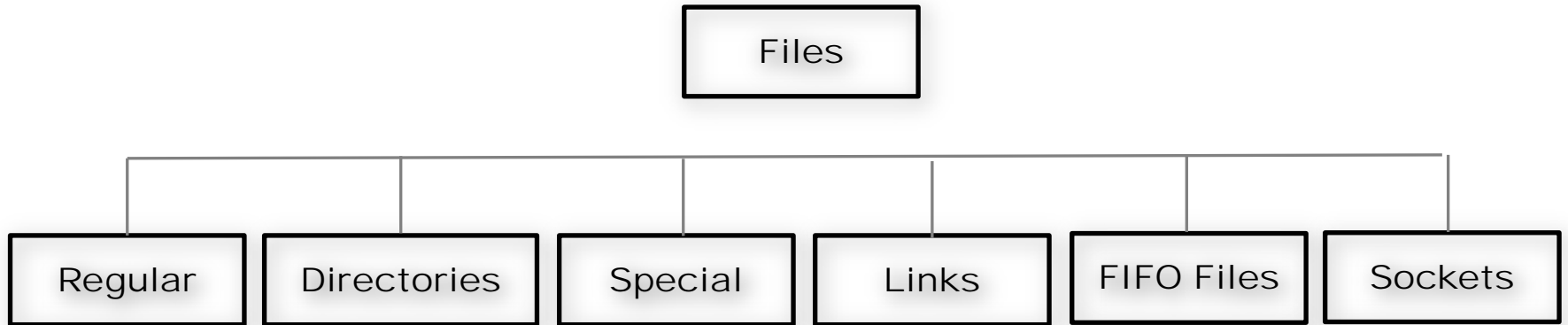
Learning Objectives

- Explain Linux filesystem
- Display and interpret directory ownership
- Change file and directory permissions
- Describe differences between hard and symbolic links.
- Describe inode, and its relationship with files and directories

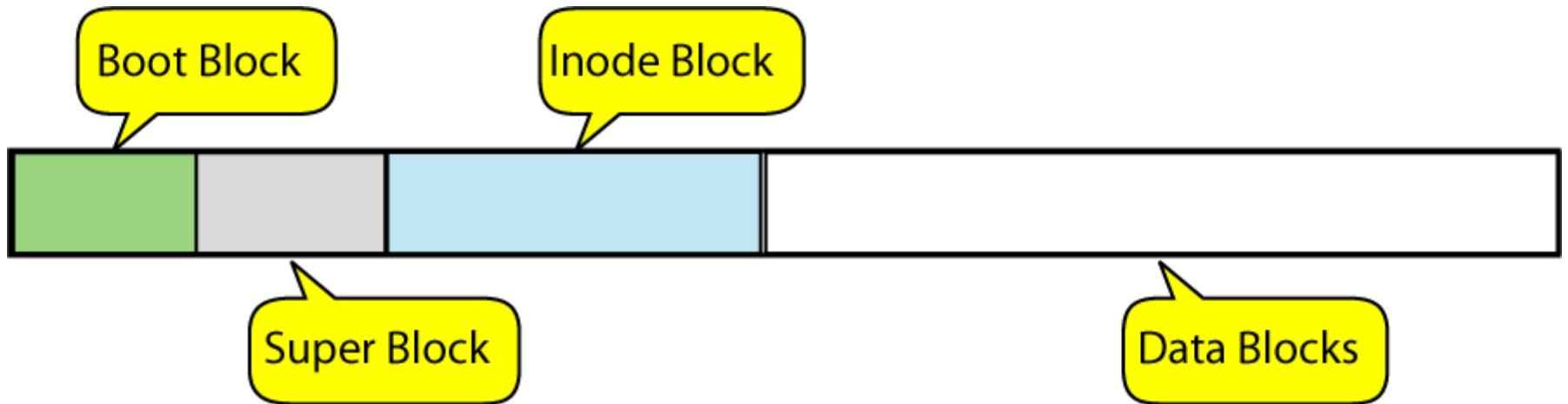
What is a File System?

- An abstraction of data storage on a system
- Used to control how information is stored and retrieved ([Wikipedia](#))
- Can be used on many different kind of media: hard drive, magnetic tape, optical disc, and flash memory

In Linux, Everything is a File!



Disk File Format



Types of Files

Directories

Regular
Files

```
[victoryu@voyager:~ ] $ls -l
total 132
drwx----- 2 victoryu victoryu 4096 Sep 16  2003 Mail
drwxr-xr-x  5 victoryu victoryu 4096 Jan 27 22:54 cis18a
drwxrwxr-x  9 victoryu victoryu 4096 Jan 20 17:51 cis18b
drwxr-xr-x  2 victoryu victoryu 4096 Jan 22 11:40 cis35a
drwxr-xr-x  3 victoryu victoryu 4096 Jul  3  2013 cis71A
-rw-r--r--  1 victoryu victoryu  193 Jan 13 23:13 data
-rw-r--r--  1 victoryu victoryu  163 Jan 13 23:13 data2
```

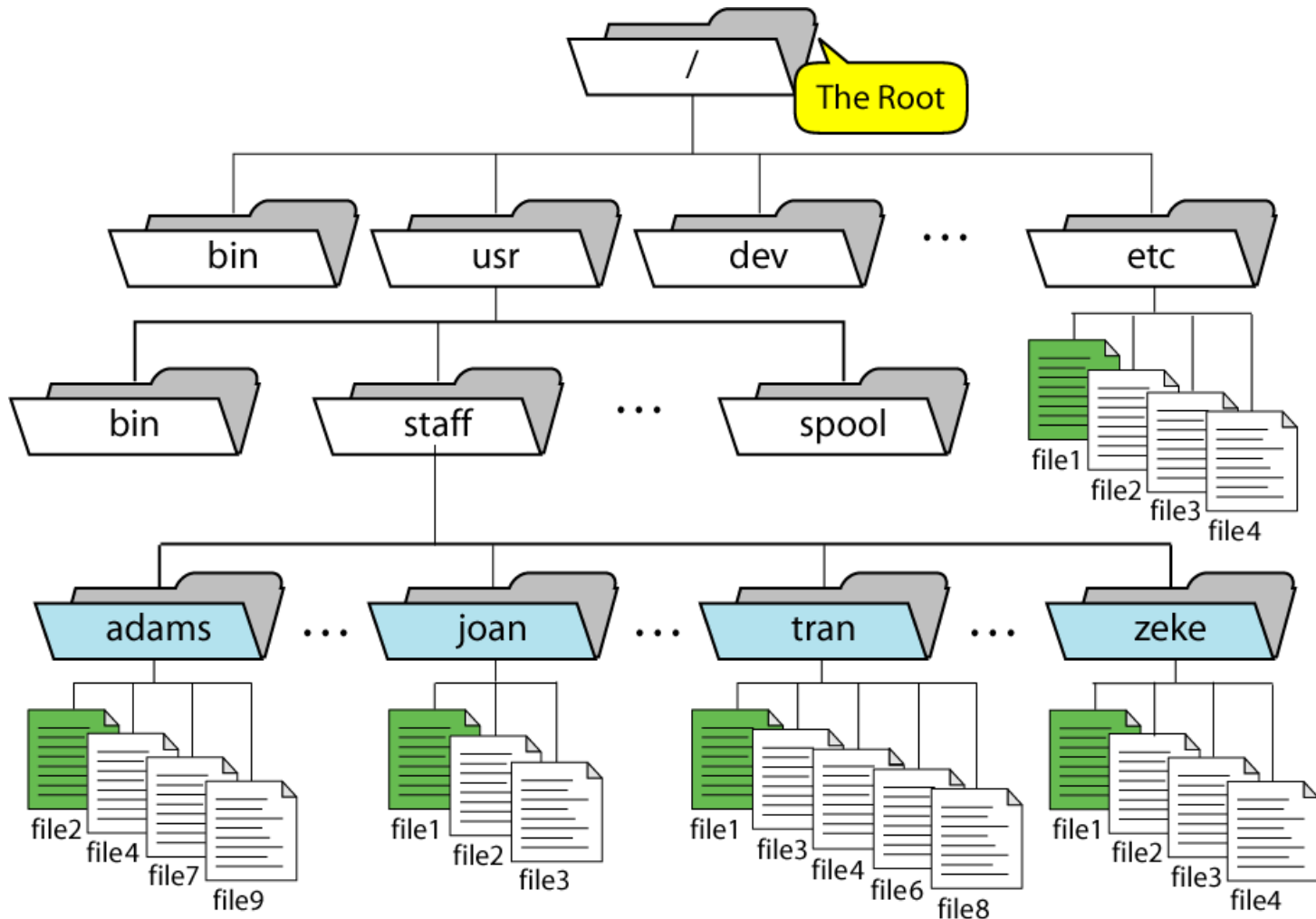
File types in a long list

Symbol	Meaning
-	Regular file
d	Directory
l	Link
c	Special file
s	Socket
p	Named pipe
b	Block device

`file` : Determine File Type

- `-b` brief
- `-z` uncompress

Hierarchical Directory Structure



Directory

Purpose

/	The root directory; all files appear in this directory or subdirectories of it.
/etc	Holds system configuration files.
/boot	Holds important boot files, such as the Linux kernel, the initial RAM disk, and often boot loader configuration files.
/bin	Holds program files that are critical for normal operation and that ordinary users may run.
/sbin	Holds program files that are critical for normal operation and that ordinary users seldom run.
/lib	Holds libraries—code used by many other programs—that are critical for basic system operation.
/usr	Holds programs and data used in normal system operation but that aren't critical for a bare-bones boot of the system. This directory is split into subdirectories that mirror parts of the root organization—/usr/bin, /usr/sbin, /usr/lib, and so on.
/home	Holds users' home directories. Separating this directory into its own low-level filesystem effectively isolates most user data from the OS, which can be useful if you want to re-install the OS without losing user data.
/root	The root user's home directory. Note that this is different from /, which is pronounced <i>root</i> .
/var	Holds miscellaneous transient files, such as log files and print spool files. One subdirectory of /var, /var/tmp, deserves special mention. Like /tmp(described next), /var/tmpholds temporary files. These files should not be deleted when the computer reboots.
/tmp	Holds temporary files, often including temporary files created by user programs. Such files may theoretically be deleted when the computer reboots, although in practice many distributions don't do this.
/mnt	The traditional mount point for removable media; sometimes split into subdirectories for each mounted filesystem.
/media	The new mount point for removable media; typically split into subdirectories for each mounted filesystem.
/dev	Holds <i>device files</i> , which provide low-level access to hardware.

File Names

- Unique name in a given directory
- Case sensitive
- Do not use SPACE in file names
- Do not use – in file names
- Hidden files

ls: Display Files

Options

`-a: list all files`

`-r: reverse order`

`--author: with -l, authors of each file`

`-R: list subdirectories recursively`

`-c: user/group ids`

`-s: print size of each file`

`-d: reverse order`

`-S: sort by file size`

`-i: list inode`

`-t: sort by modification time`

`-l: long list format`

`-u: show access time`

`-L: info for link references`

`-1: show one file per line`

The touch Command

- Update file time access time
- Create an empty file

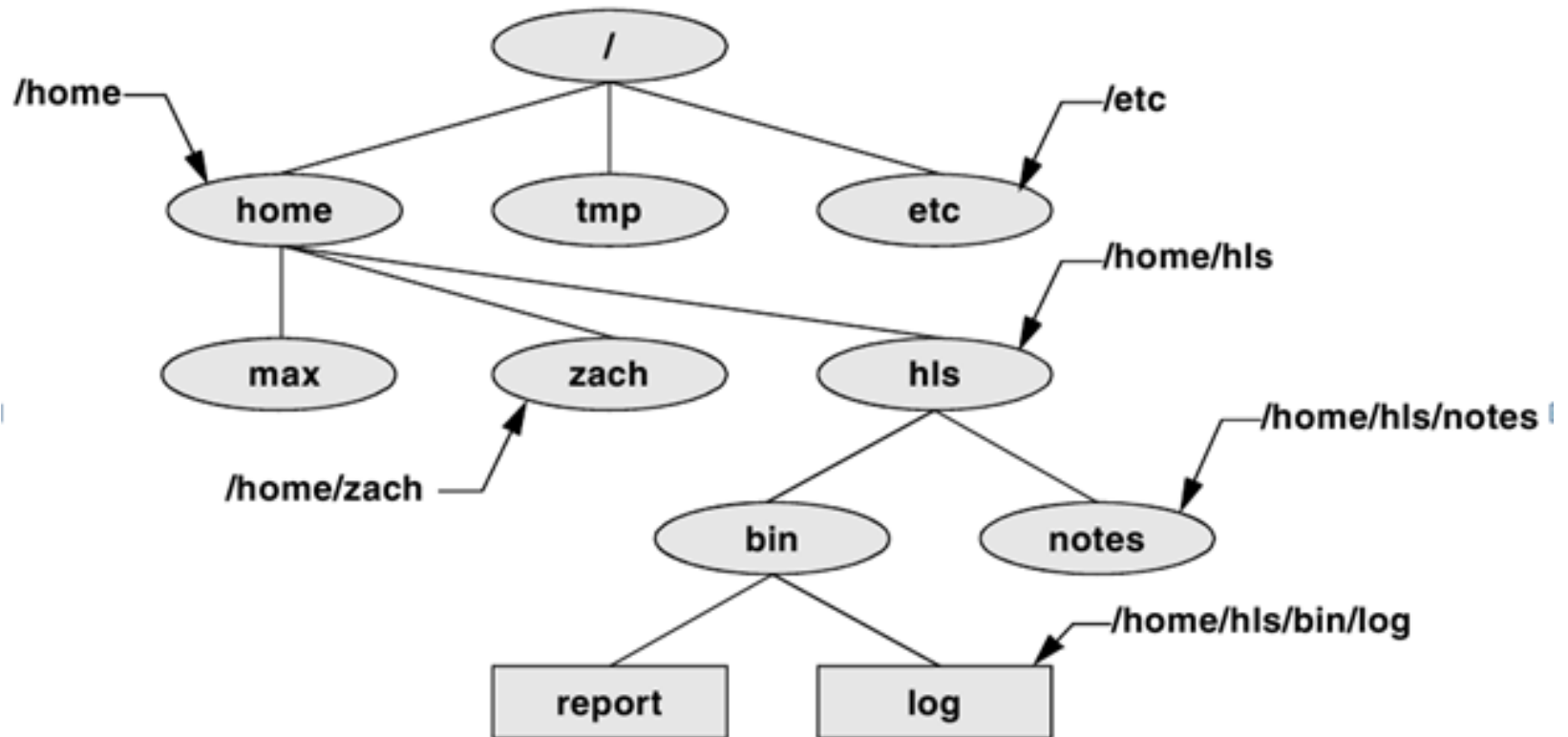
mkdir: Create a Directory

```
$ pwd
/home/max
$ ls
demo  names  temp
$ mkdir literature
$ ls
demo  literature  names  temp
$ ls -F
demo  literature/  names  temp
$ ls literature
$
```

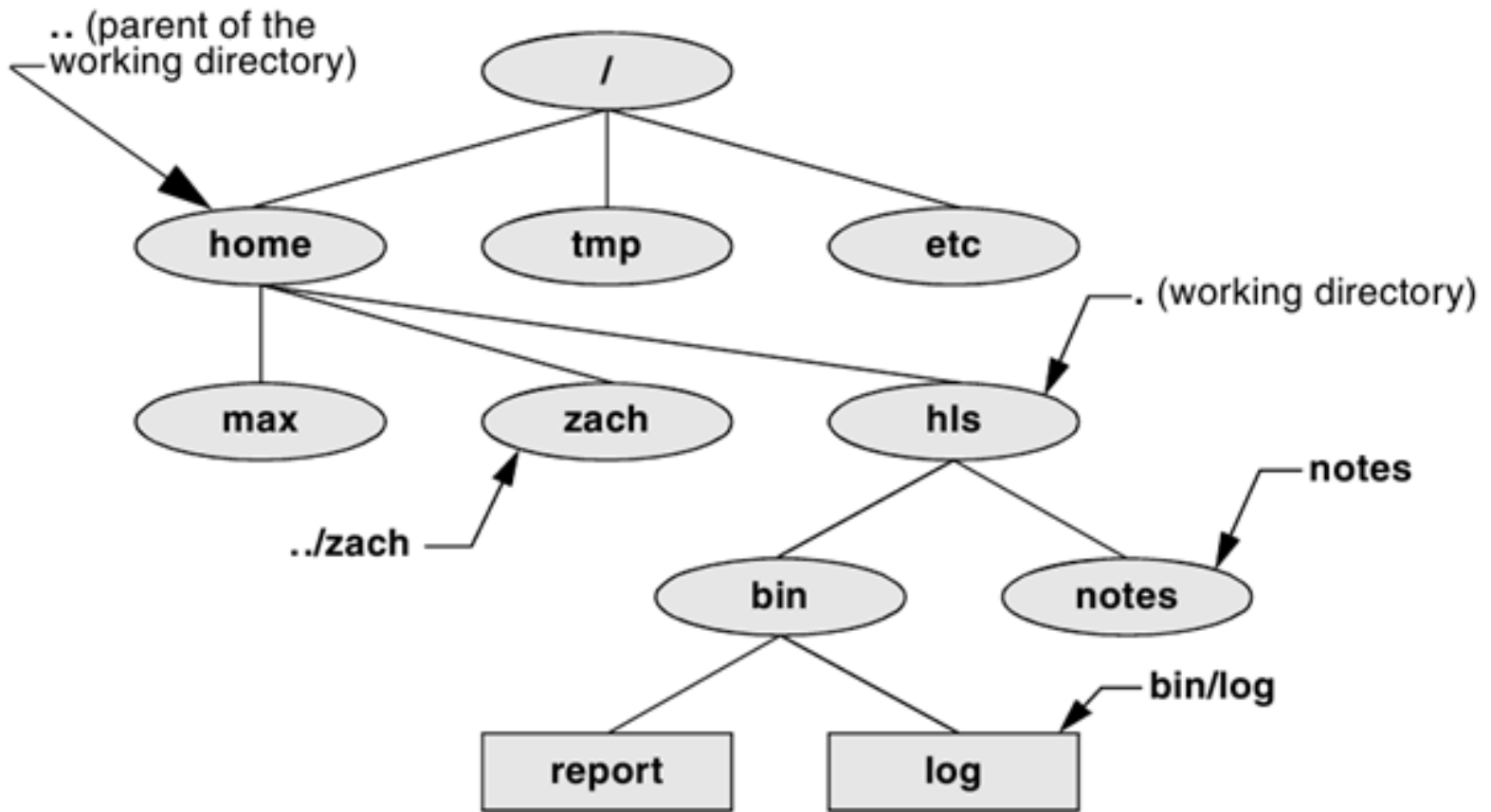
Paths

- Absolute paths
 - / (root) - root directory
 - ~ (tilder) – home directory
- Relative paths
 - . . - parent directory
 - . – current directory

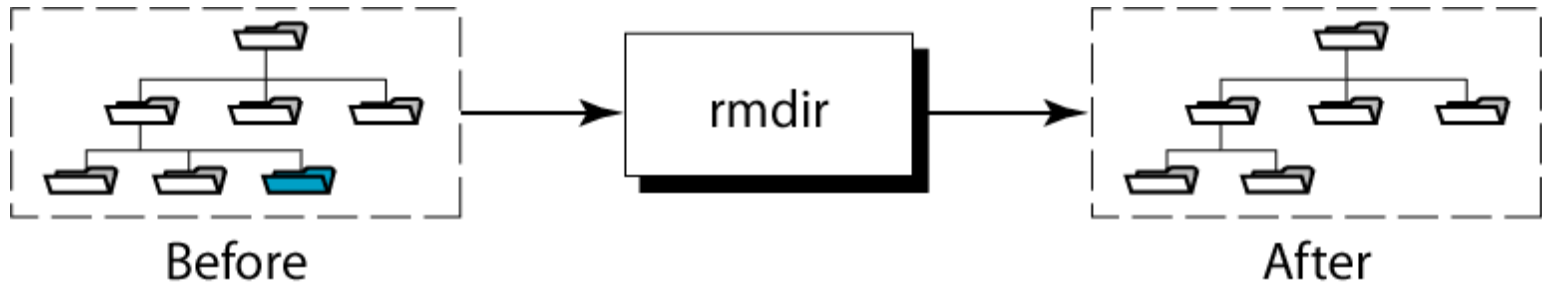
Absolute Paths



Relative Paths

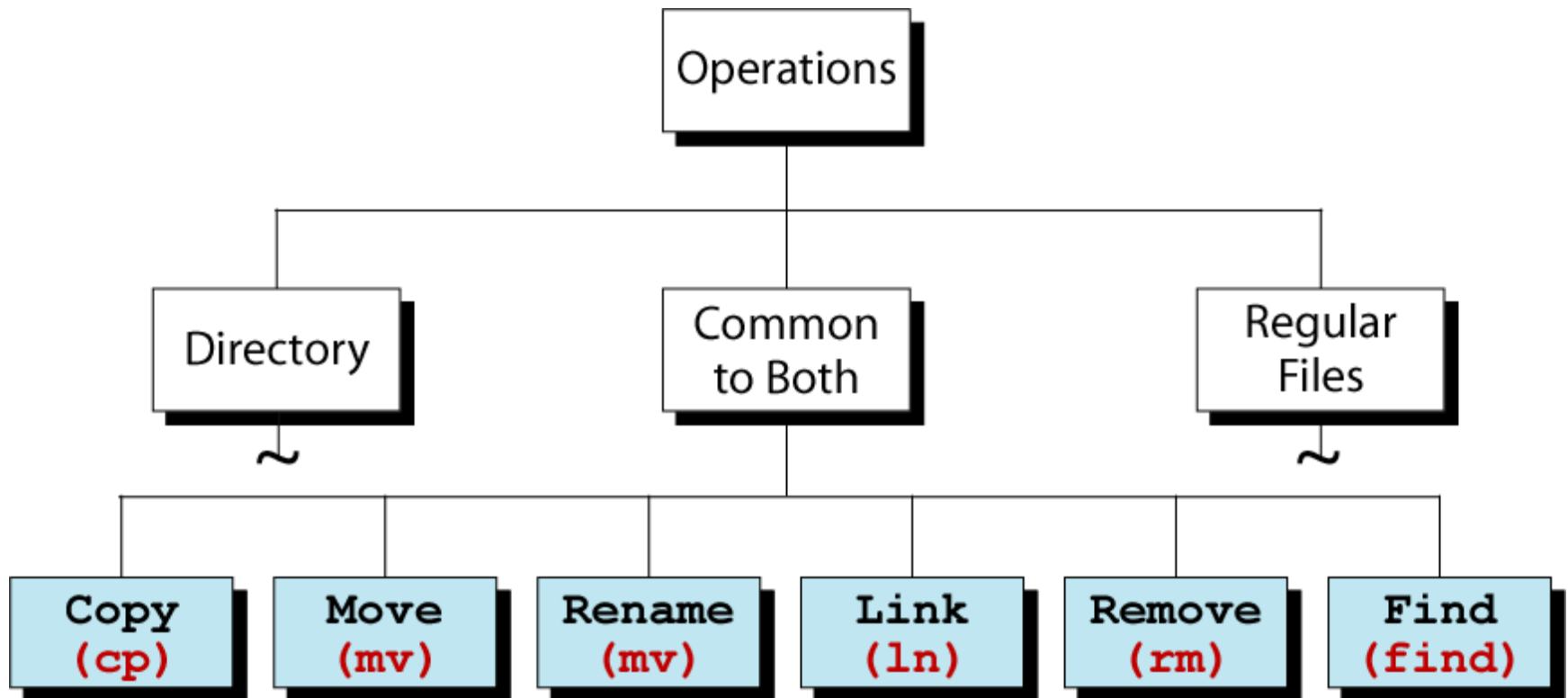


`rmdir`: Remove a Directory

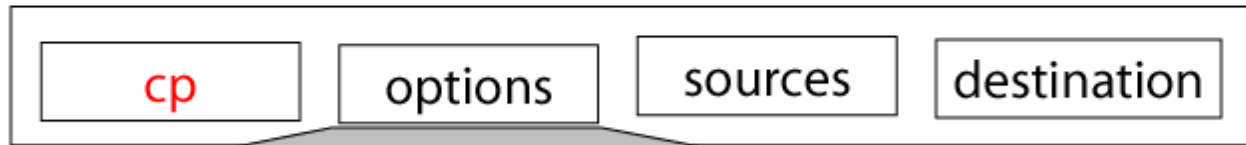
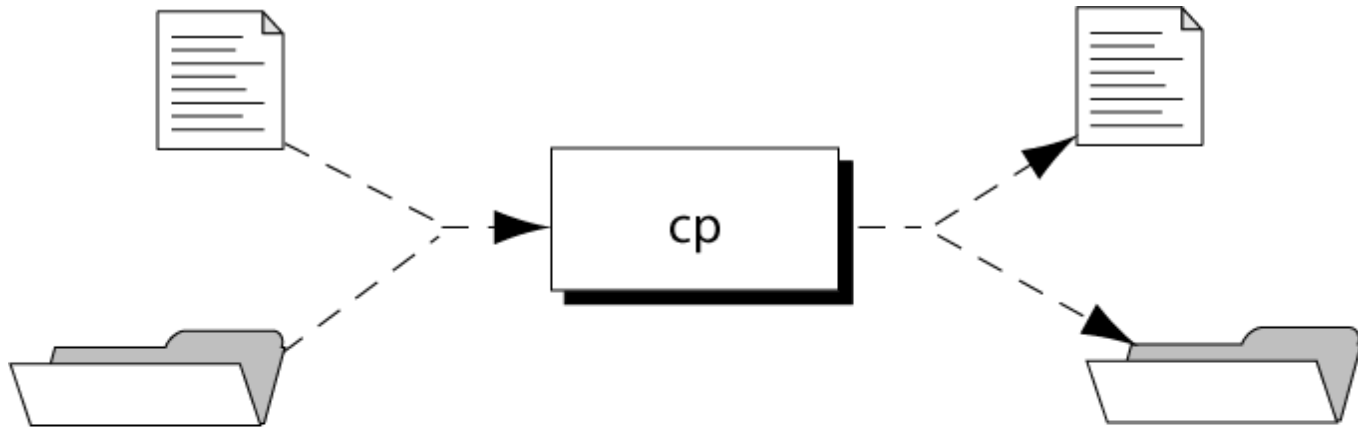


<code>rmdir</code>	<code>-options</code>	<code>dir name</code>
--------------------	-----------------------	-----------------------

Regular File Utilities



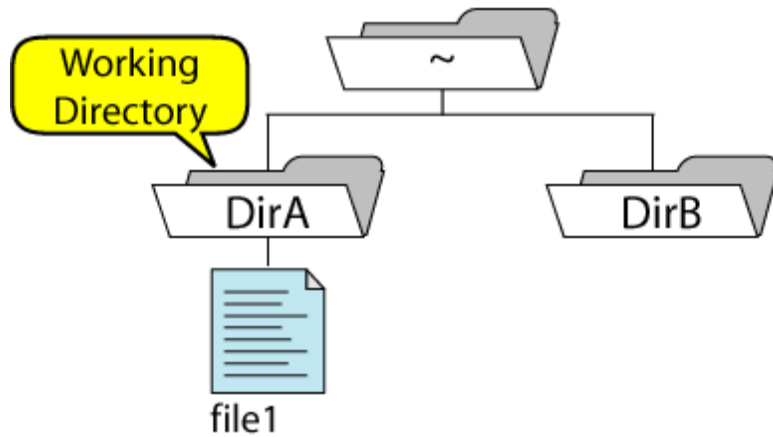
The cp Command



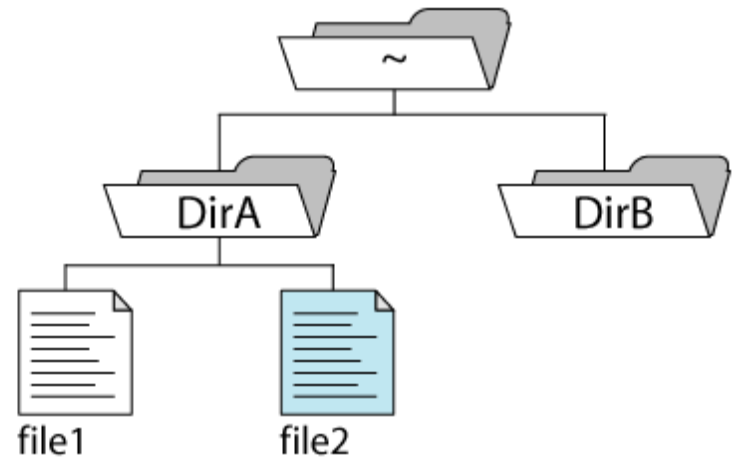
-p: preserve time and permissions
-i: interactive prompt
-r: recursive—copy subdirectories

Simple File Copy

```
$ cp file1 file2
```

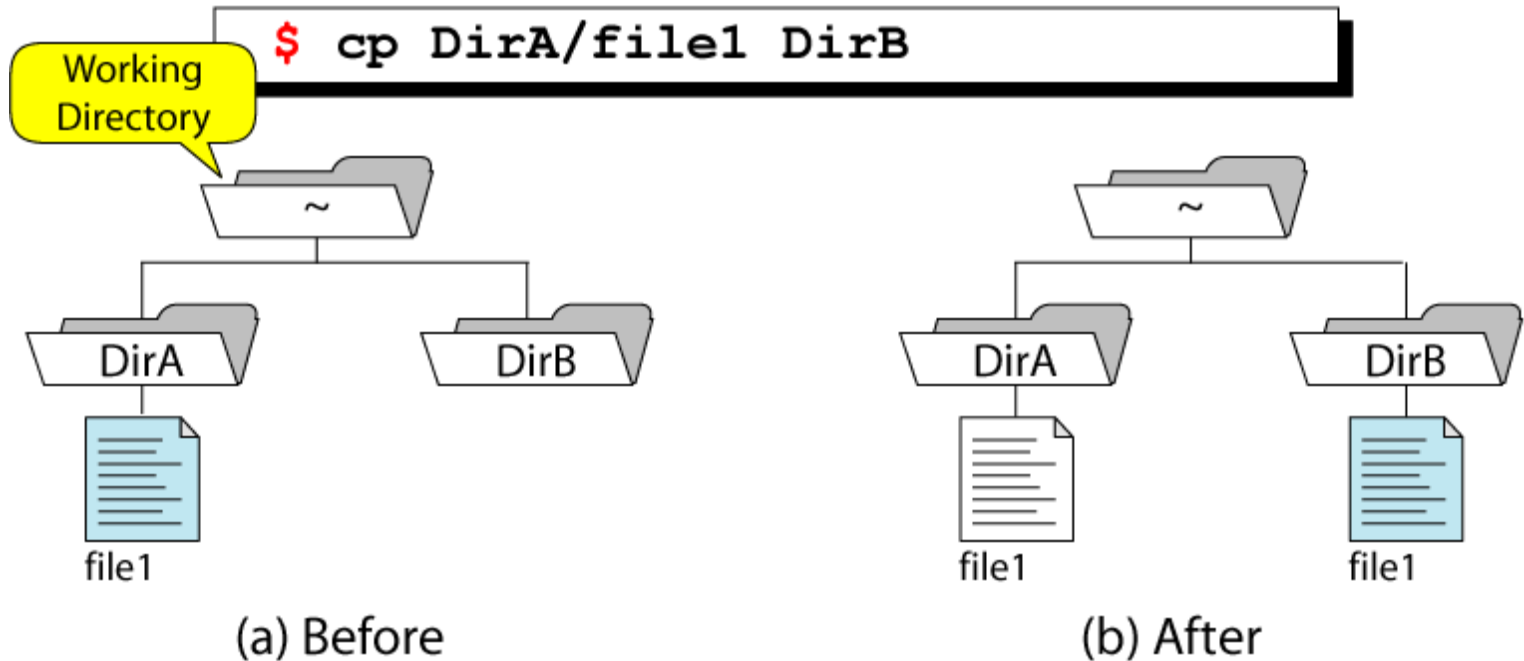


(a) Before



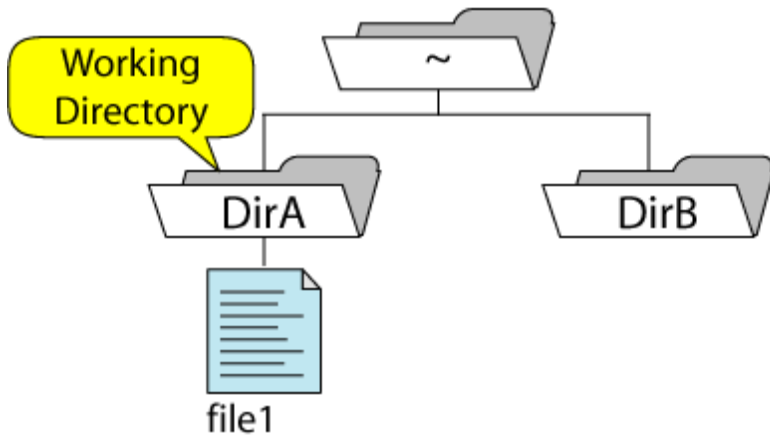
(b) After

Copy File to a Directory from Home

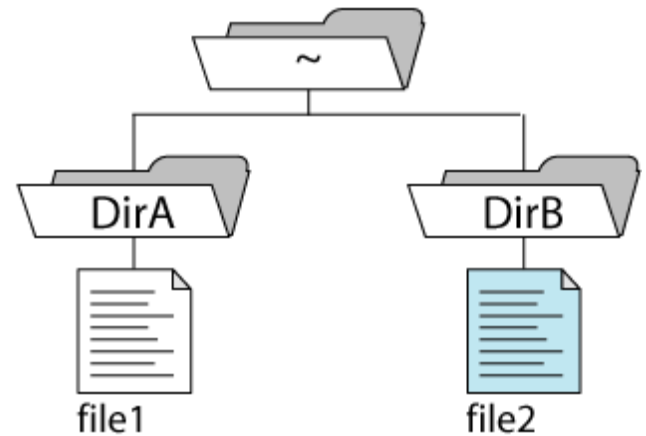


Copy and Rename a File

```
$ cp file1 ~/DirB/file2
```

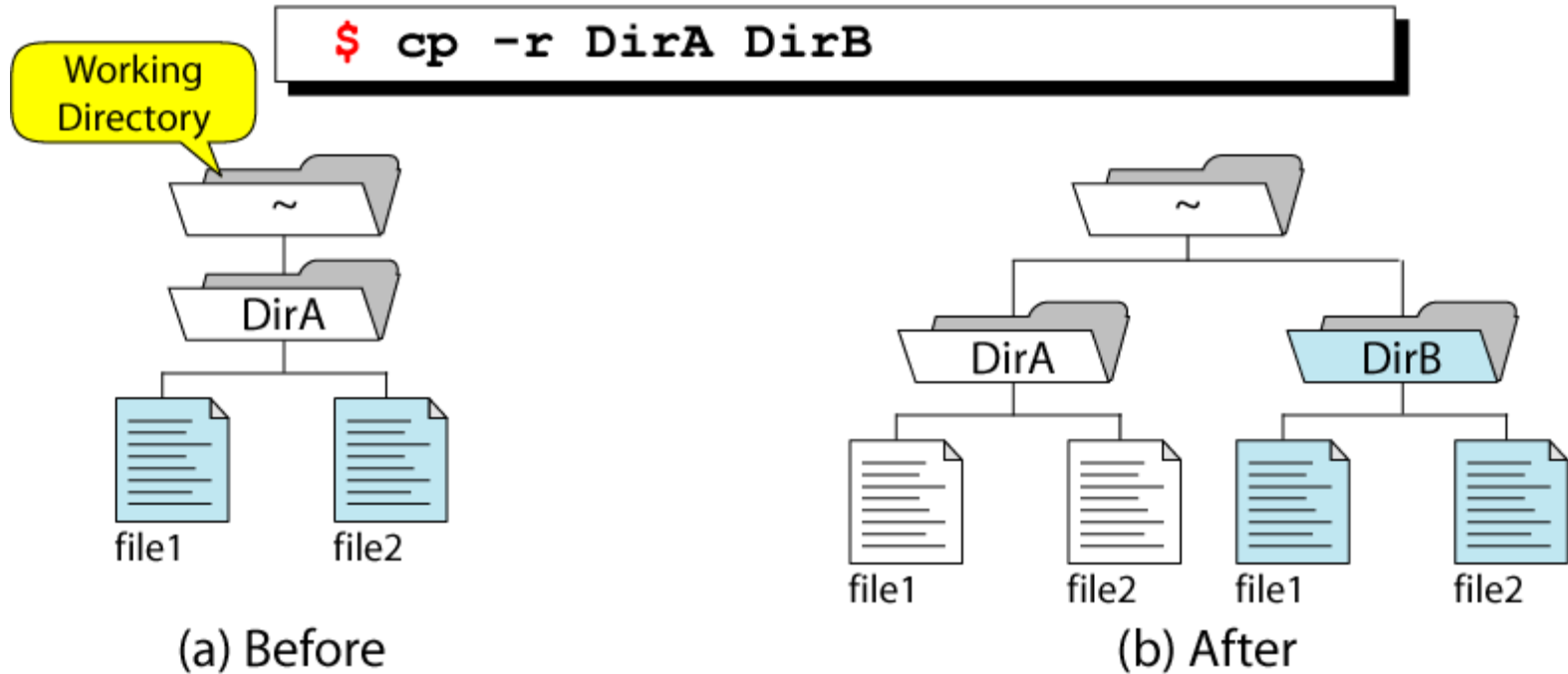


(a) Before

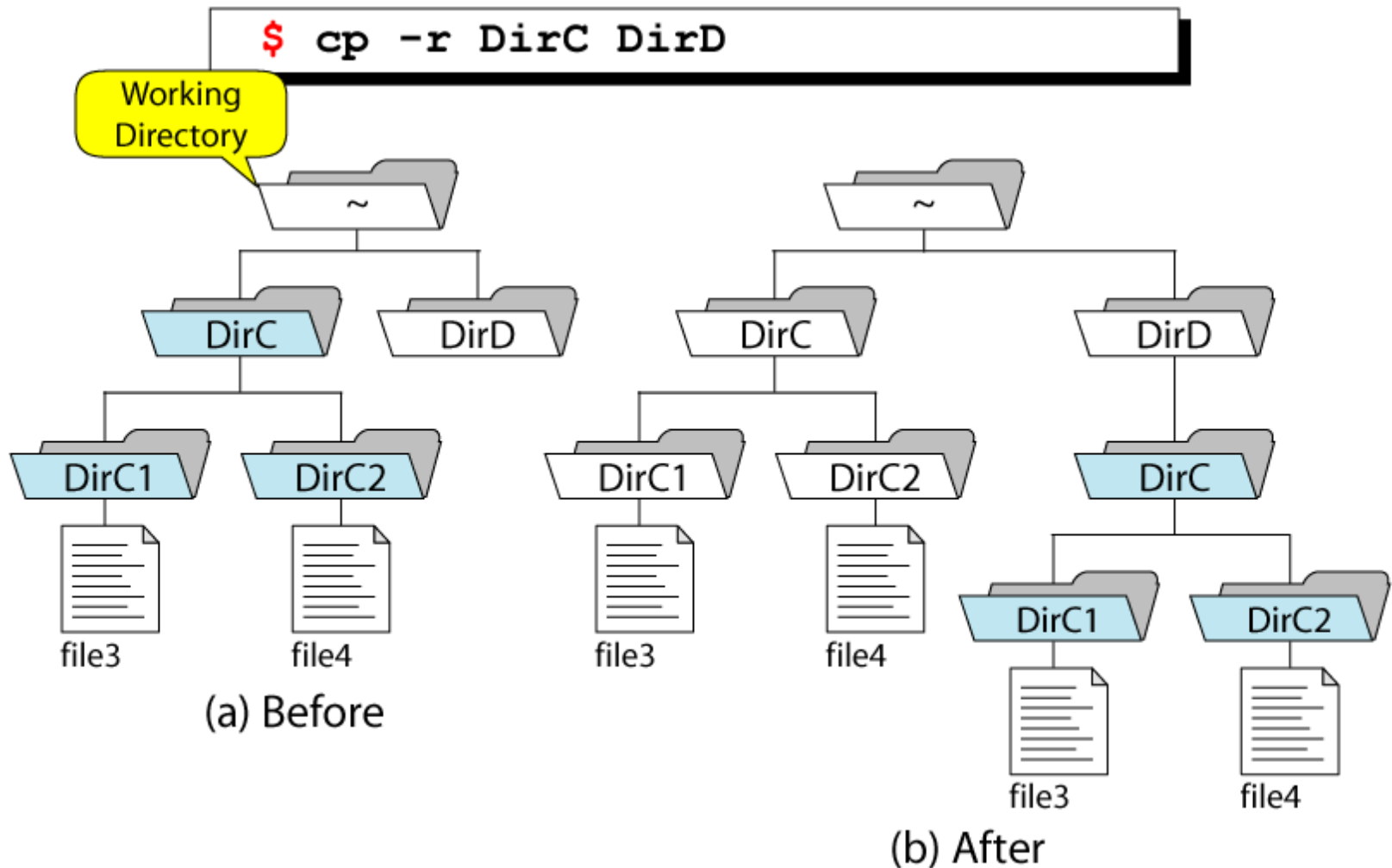


(b) After

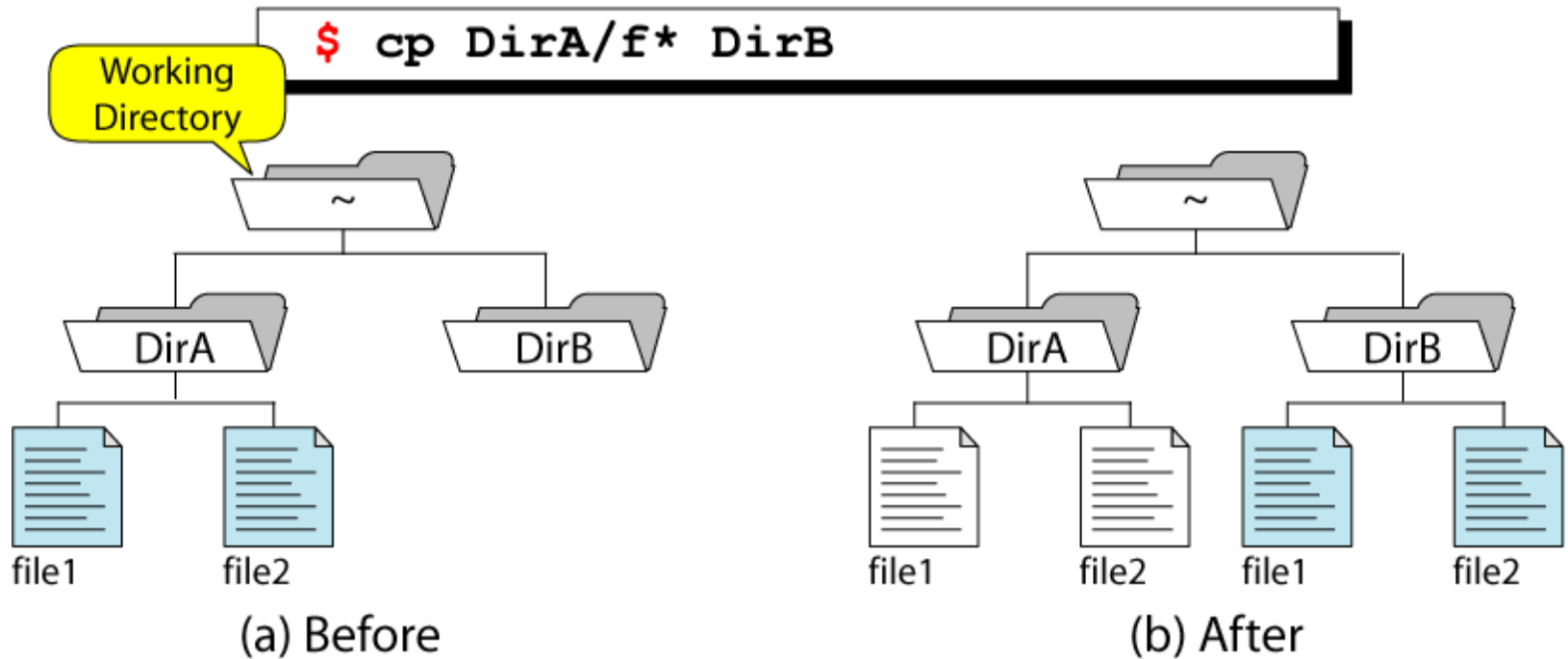
Recursive Copy



Recursive Copy with Subdirectories



Wildcard Copy



File Ownership and Permissions

- Three levels of file ownership
 - User
 - Group
 - Other
- Within each level, 3 types of permissions
 - Read
 - Write
 - Execute

ls -l: Displays Permissions, etc.

Type of file	File access permissions	ACL flag	Links	Owner	Group	Size	Date (and time) of modification	Filename
-	<u>rw</u> xr-xr-x+		1	sam	pubs	2048	<u>06-10 10:44</u>	memo

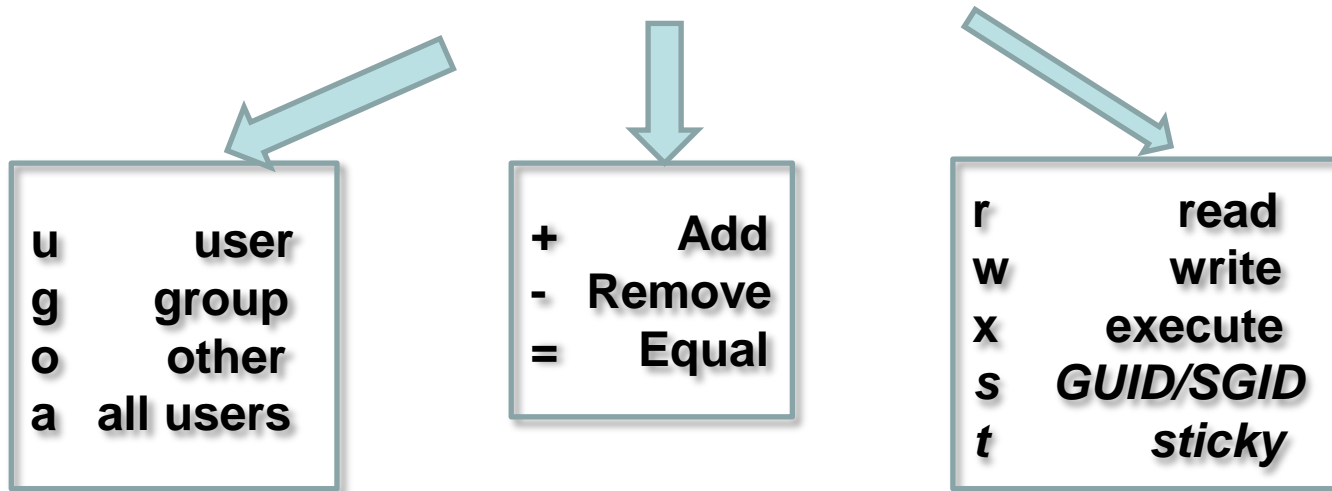
chown: Change Ownership

- After the change, the new owner will be able to do anything with the file
- Use `-R` option to apply recursively
- If the system does not allow you to use `chown`, consider using `cp` to accomplish the same thing.

Symbolic Arguments to chmod

Choose One From Each

User Operator Permission



```
[victoryu@voyager:~/cis18a ] $ls -l testfile
-rw-r--r-- 1 victoryu victoryu 22 Feb  3 21:36 testfile
[victoryu@voyager:~/cis18a ] $chmod u+rwx,g+w,o+x testfile
[victoryu@voyager:~/cis18a ] $ls -l testfile
-rwxrw-r-x 1 victoryu victoryu 22 Feb  3 21:36 testfile
[victoryu@voyager:~/cis18a ] $chmod g-r,o+x testfile
[victoryu@voyager:~/cis18a ] $ls -l testfile
-rwx-w-r-x 1 victoryu victoryu 22 Feb  3 21:36 testfile
```

Absolute Arguments to chmod

```
$ls -l | cut -d" "
```

```
$chmod 270 test2$ls -l testfile1 | cut -d" " -f1
```

```
-rwxrw-r-x
```

```
$chmod 765 testfile
```

```
-rwxrw-r-x
```

Numeric Equivalent for Mnemonic Permissions

Mnemonic (rwx)	Binary	Octal
---	000	0
--x	001	1
-w-	010	2
-wx	011	3
r--	100	4
r-x	101	5
rw-	110	6
rwX	111	7

User Mask

Mask	Directory Permission (Default 777)	File Permission (Default 666)
0	7 (rwx)	6 (rw-)
1	6 (rw-)	6 (rw-)
2	5 (r-x)	4 (r- -)
3	4 (r- -)	4 (r- -)
4	3 (-wx)	2 (-w-)
5	2 (-w-)	2 (-w-)
6	1 (- -x)	0 (- - -)
7	0 (- - -)	0 (- - -)

umask Utility

```
$umask
```

```
000
```

```
$umask 022
```

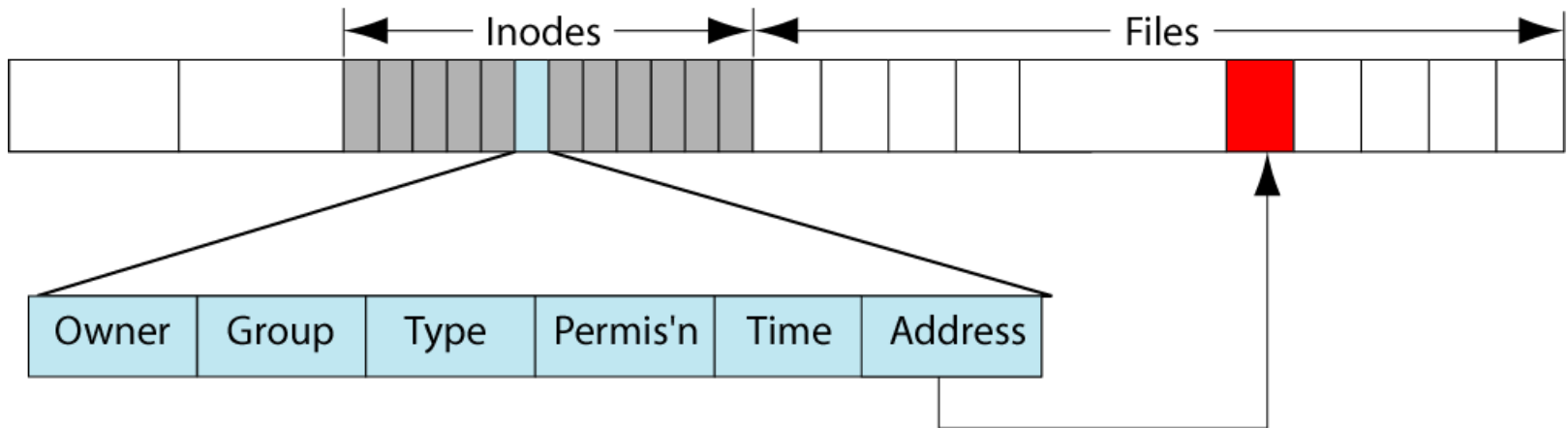
Examples of Default Permission Calculation

Mask	Directory Permissions (Default 777)	File Permissions (Default 666)
000 (Public)	777 (rwxrwxrwx)	666 (rw-rw-rw)
011 (Public)	766 (rwxrw-rw-)	666 (rw-rw-rw)
022 (Write Protected)	755 (rwxr-xr-x)	644 (rw-r-r--)
007 (Project Private)	770 (rwxrwx---)	660 (rw-rw---)
077 (Private)	700 (rwx-----)	600 (rw-----)

hostname: Displays the System Name

```
[victoryu@voyager:~/cis18a ] $hostname
voyager.deanza.edu
[victoryu@voyager:~/cis18a ] $hostname -a
voyager
[victoryu@voyager:~/cis18a ] $hostname -d
deanza.edu
[victoryu@voyager:~/cis18a ] $hostname -f
voyager.deanza.edu
[victoryu@voyager:~/cis18a ] $hostname -i
153.18.17.12
[victoryu@voyager:~/cis18a ] $hostname -s
voyager
[victoryu@voyager:~/cis18a ] $hostname -V
net-tools 1.60
hostname 1.100 (2001-04-14)
[victoryu@voyager:~/cis18a ] $hostname -y
(none)
[victoryu@voyager:~/cis18a ] $
```

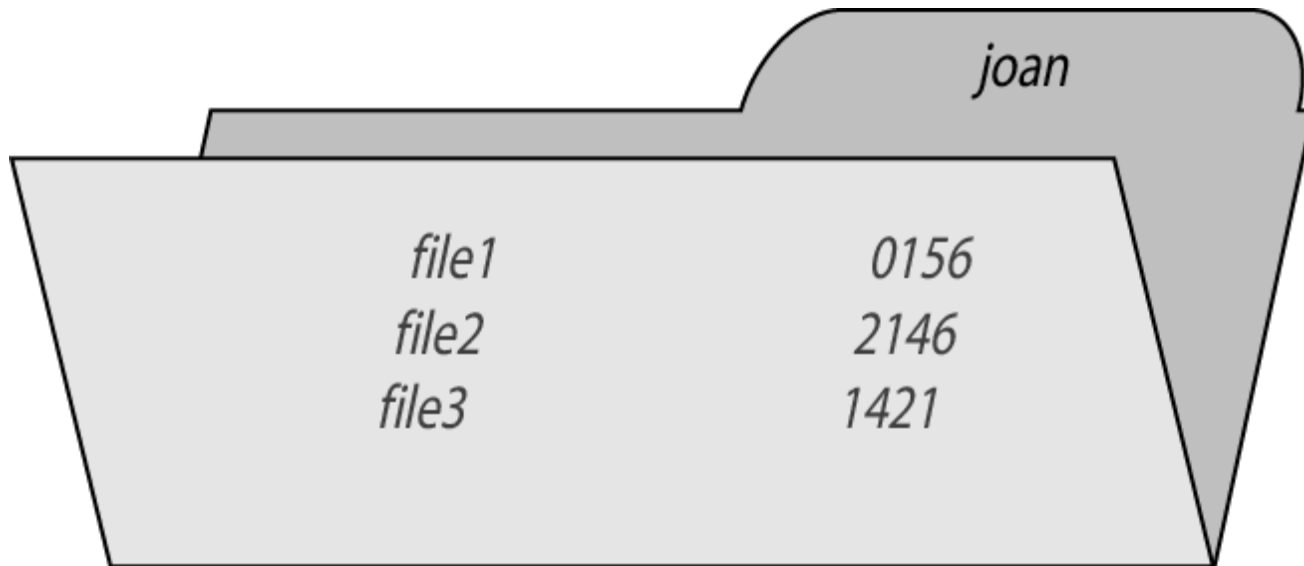
Inode



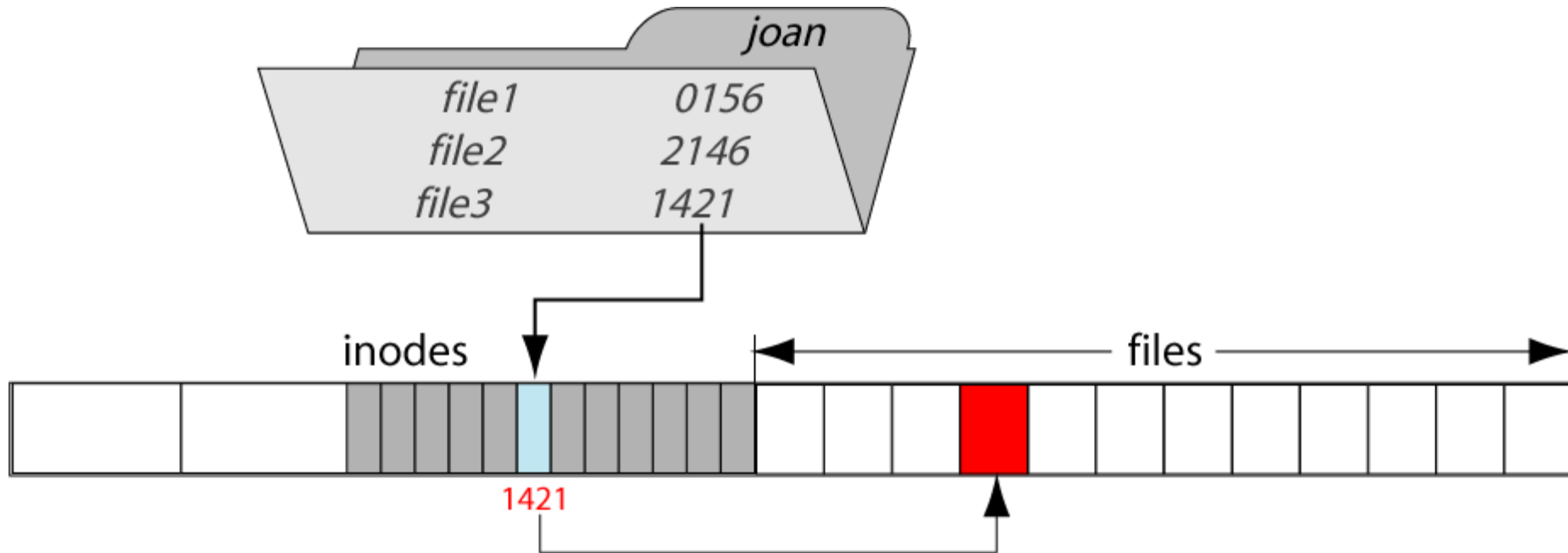
Links

- A pointer to a file
 - Hard link – the specific location of physical data
 - Symbolic (or soft) links – pointers to a
- Two pieces of information
 - A name
 - An inode number

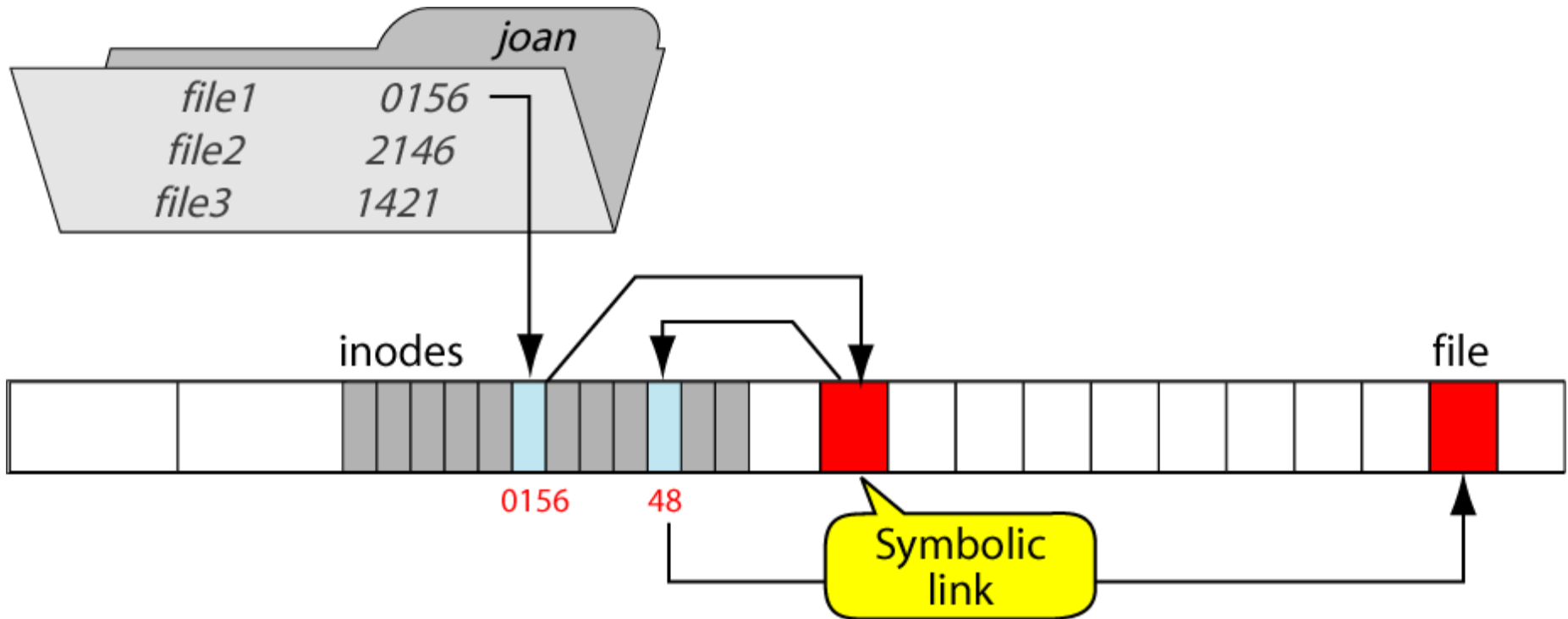
A Directory of Three Files



A Hard Link



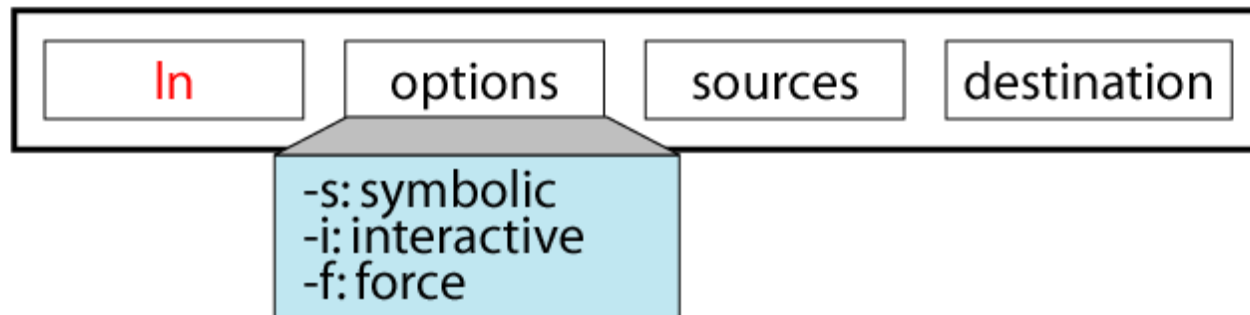
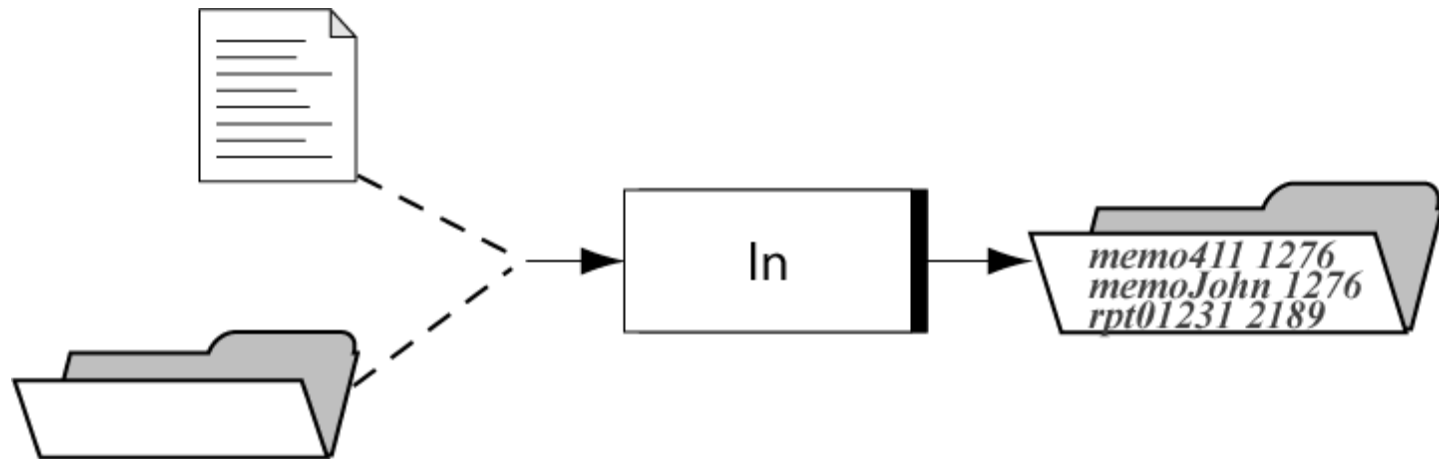
A Symbolic Link



Hard Links vs. Soft Links

- A hard link is an alias to to a file
 - Hard link – the specific location of physical data
 - Can't make a hard link to a directory
 - Can 't span file systems
- A symbolic link is a pointer to a pathname, not a pointer to the file itself
 - `ln -s original target` creates a symbolic link
 - Is not equivalent to a hard link, and has a different inode
 - Can make a soft link to a directory

The ln Command



1n: Create Link

```
[victoryu@voyager:~/cis18a/testDir ] $ls
thursday tuesday
[victoryu@voyager:~/cis18a/testDir ] $ln tuesday mardi
[victoryu@voyager:~/cis18a/testDir ] $ls
mardi thursday tuesday
[victoryu@voyager:~/cis18a/testDir ] $ln -s thursday jeudi
[victoryu@voyager:~/cis18a/testDir ] $ls
jeudi mardi thursday tuesday
[victoryu@voyager:~/cis18a/testDir ] $ls -l
total 28
lrwxrwxrwx 1 victoryu victoryu 8 Feb 3 16:04 jeudi -> thursday
-rw-r--r-- 2 victoryu victoryu 17 Feb 3 16:02 mardi
-rw-r--r-- 1 victoryu victoryu 18 Feb 3 16:03 thursday
-rw-r--r-- 2 victoryu victoryu 17 Feb 3 16:02 tuesday
[victoryu@voyager:~/cis18a/testDir ] $
[victoryu@voyager:~/cis18a/testDir ] $
```

find : Search for a File

- Syntax: `find [path] [option] filename`

- `-name` File name

```
find . -name "Jose*"
```

```
find /home -name test.txt
```

- `-perm` File permission

```
find . -perm 644
```

- `-atime n` File status last accessed n*24 hours ago

- `-empty` File is empty

- `find [path] [action] command`

- `-exec`

```
find . -name file3 -exec cp {} {}.bak \; -exec rm {} \;
```

Command Comparison

Command	Apply to Files?	Apply to Directories?
<code>cat</code>	Yes	No
<code>cd</code>	No	Yes
<code>rm</code>	Yes	Yes
<code>cp</code>	Yes	Yes
<code>mv</code>	Yes	Yes
<code>ls</code>	Yes	Yes
<code>find</code>	Yes	Yes
<code>mkdir</code>	No	Yes
<code>rmdir</code>	No	Yes (empty directory only)
<code>touch</code>	Yes	Yes

ACLs: Access Control Lists

- ACLs provide a set of finer-grained rules specifying how a specific user or group can access a file or directory

- Syntax:

```
setfacl -modify ugo:name:permission file
```

- Example: set default rules

```
setfacl -d -m g:pubs:r-x,g:adm:rwX dir
```

- Example: modify rule(s) in a file's ACL

```
setfacl -modify g:unixStudent:rw dropbox
```

- Example: display info about an ACL associated file

```
getfacl dropbox
```

Review Questions

- What is an inode? What happens to the inode when you move a file within a filesystem?
- How can you create a file named -i? Which techniques do not work, and why don't they work? How can you remove the file name -i?
- How to find all files with permission 644 and SGID bit set?