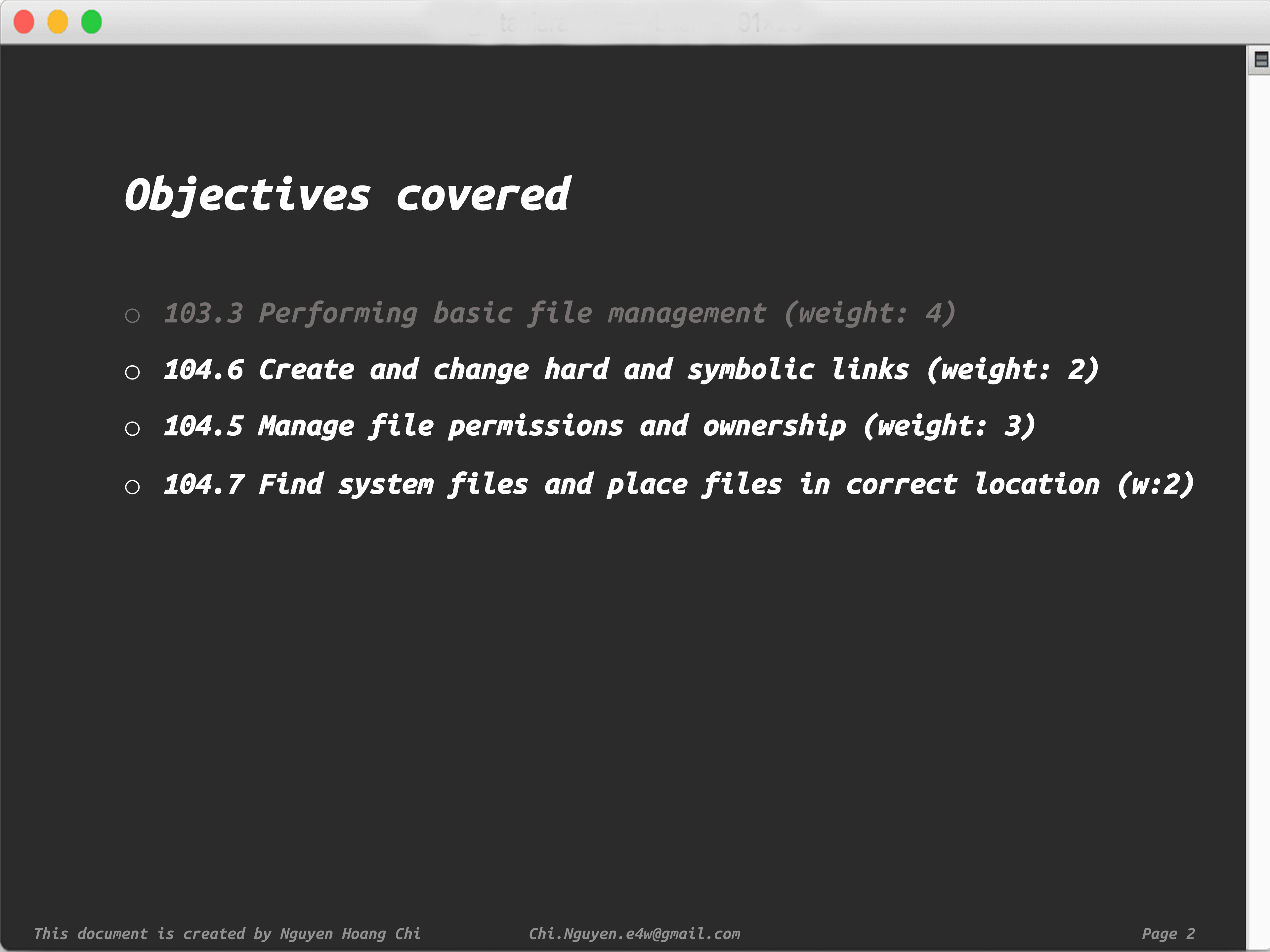
Lesson 5: Managing Files

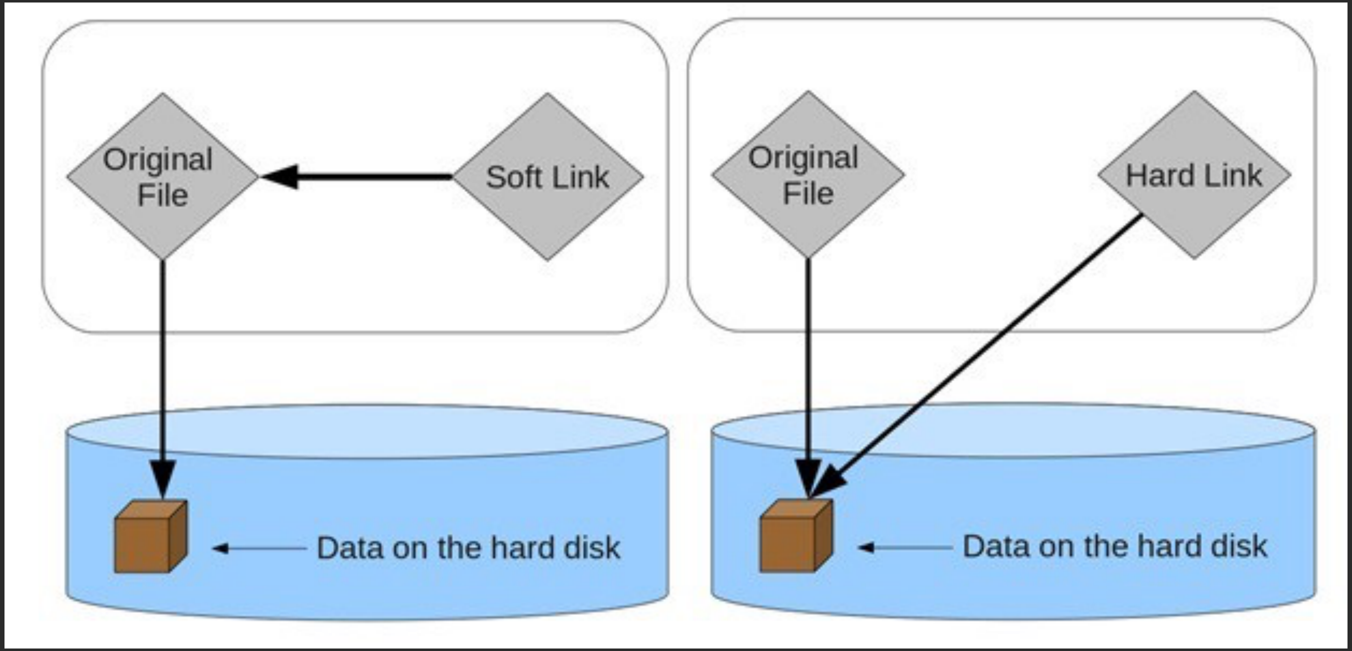


Objectives covered

- ○ 103.3 Performing basic file management (weight: 4)
- ○ 104.6 Create and change hard and symbolic links (weight: 2)
- ○ 104.5 Manage file permissions and ownership (weight: 3)
- ○ 104.7 Find system files and place files in correct location (w:2)

Create and change hard and symbolic links

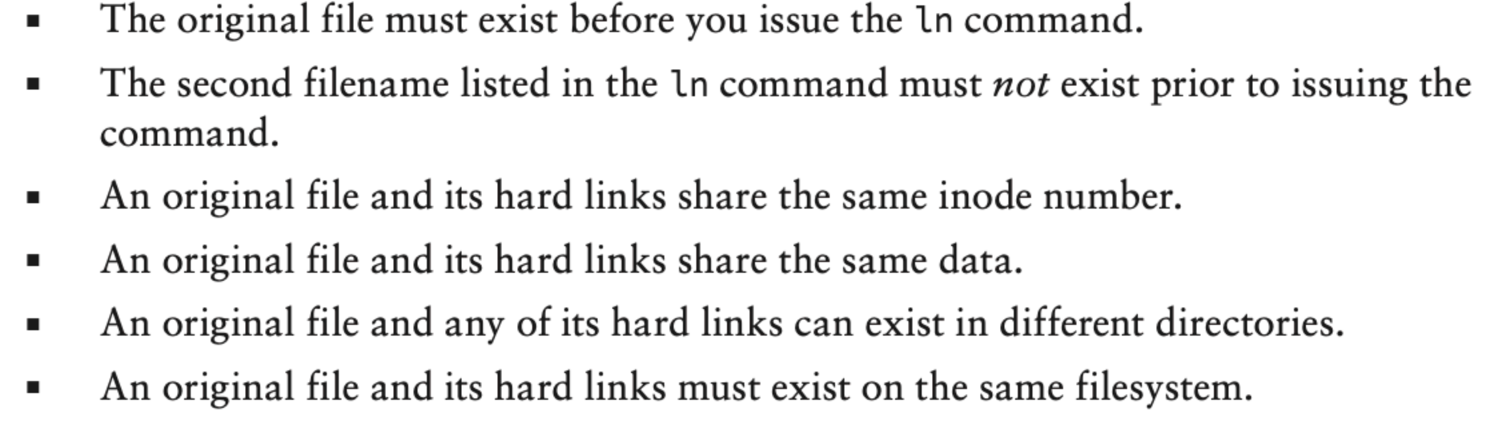
Hardlink vs Softlink



Creating Hardlink

```
ln file link
```

Notes:

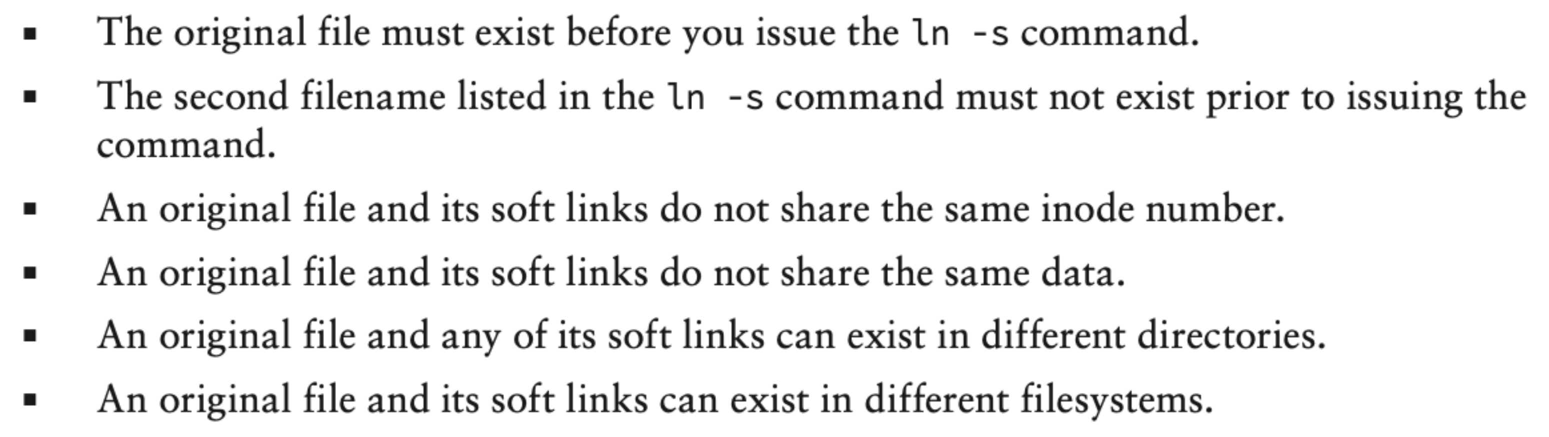


- The original file must exist before you issue the `ln` command.
- The second filename listed in the `ln` command must *not* exist prior to issuing the command.
- An original file and its hard links share the same inode number.
- An original file and its hard links share the same data.
- An original file and any of its hard links can exist in different directories.
- An original file and its hard links must exist on the same filesystem.

Creating Softlink

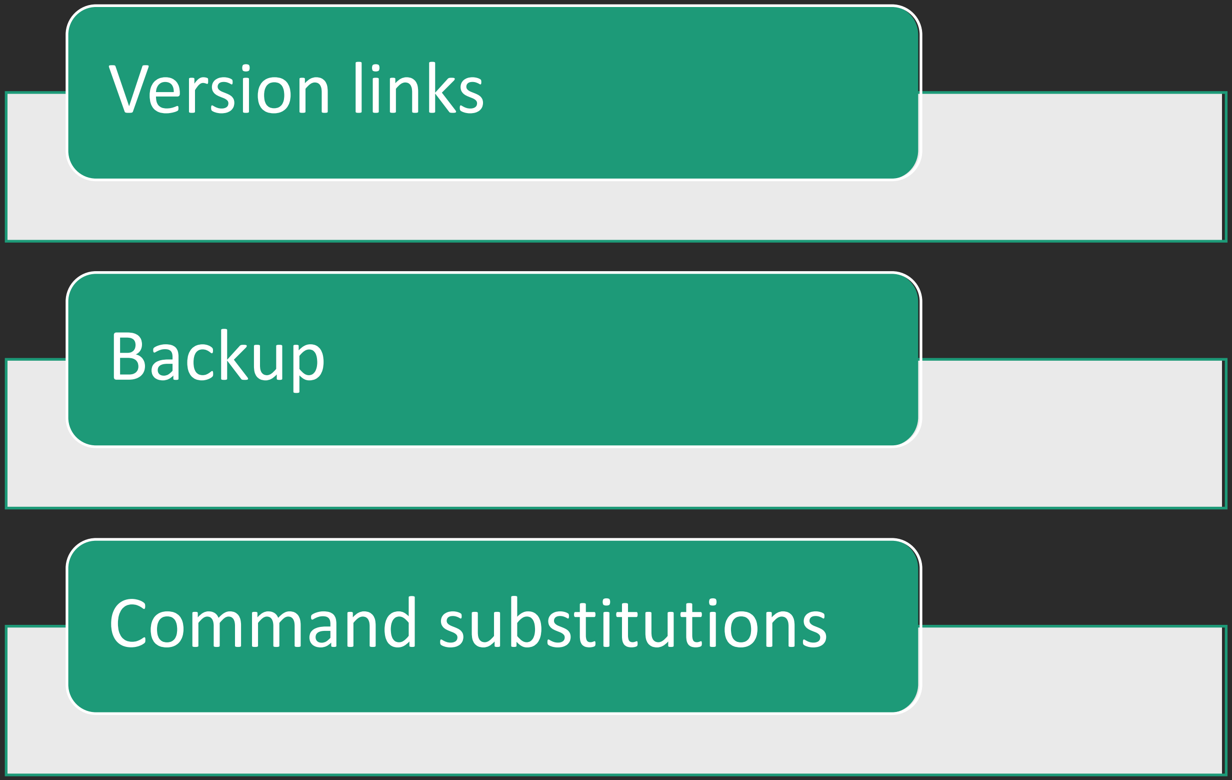
```
ln -s file link
```

Notes:



- The original file must exist before you issue the `ln -s` command.
- The second filename listed in the `ln -s` command must not exist prior to issuing the command.
- An original file and its soft links do not share the same inode number.
- An original file and its soft links do not share the same data.
- An original file and any of its soft links can exist in different directories.
- An original file and its soft links can exist in different filesystems.

Link usages



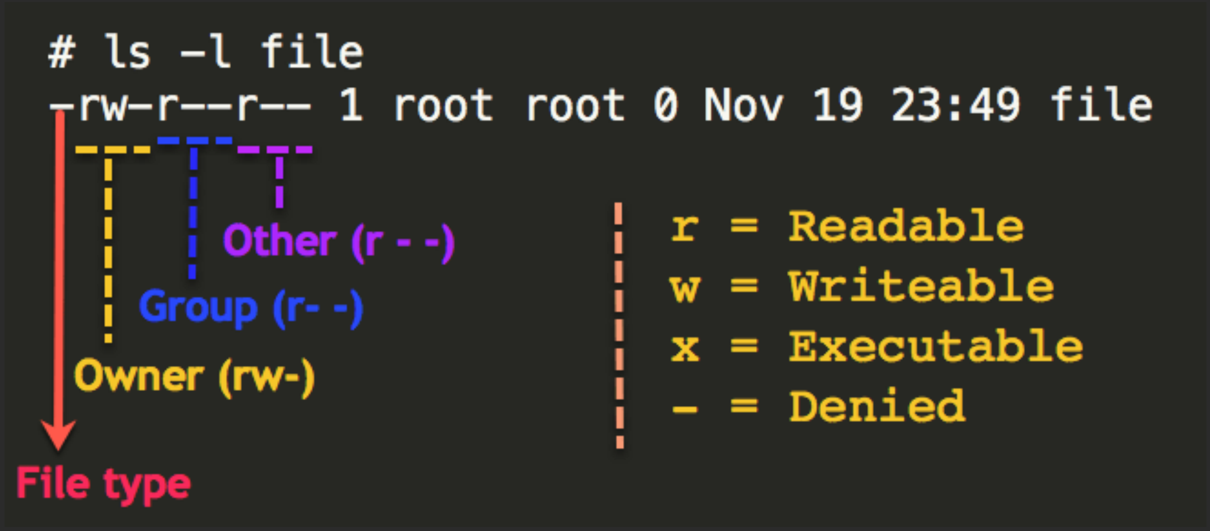
Version links

Backup

Command substitutions

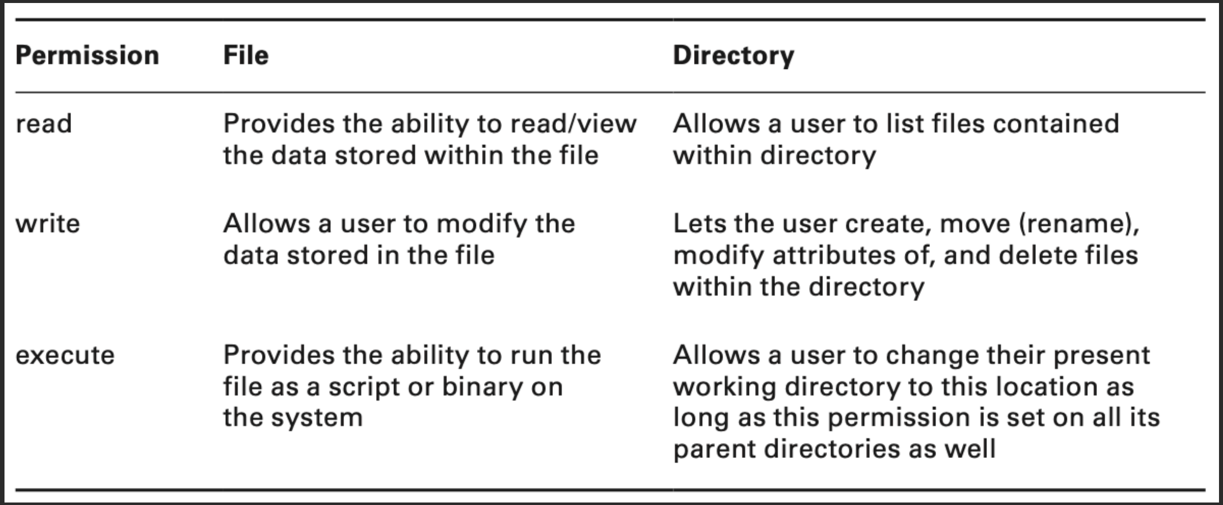
Manage file permissions and ownership

Basic file permissions



```
# ls -l file
-rw-r--r-- 1 root root 0 Nov 19 23:49 file
```

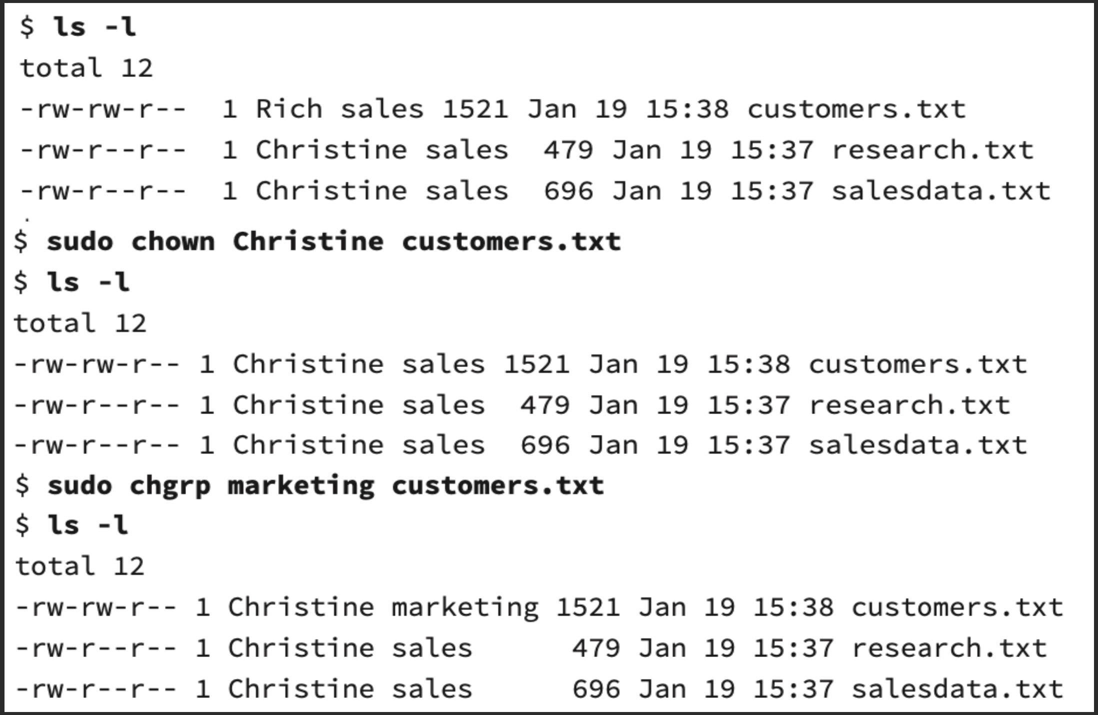
```
r = Readable
w = Writeable
x = Executable
- = Denied
```



Permission	File	Directory
read	Provides the ability to read/view the data stored within the file	Allows a user to list files contained within directory
write	Allows a user to modify the data stored in the file	Lets the user create, move (rename), modify attributes of, and delete files within the directory
execute	Provides the ability to run the file as a script or binary on the system	Allows a user to change their present working directory to this location as long as this permission is set on all its parent directories as well

Change owner and group

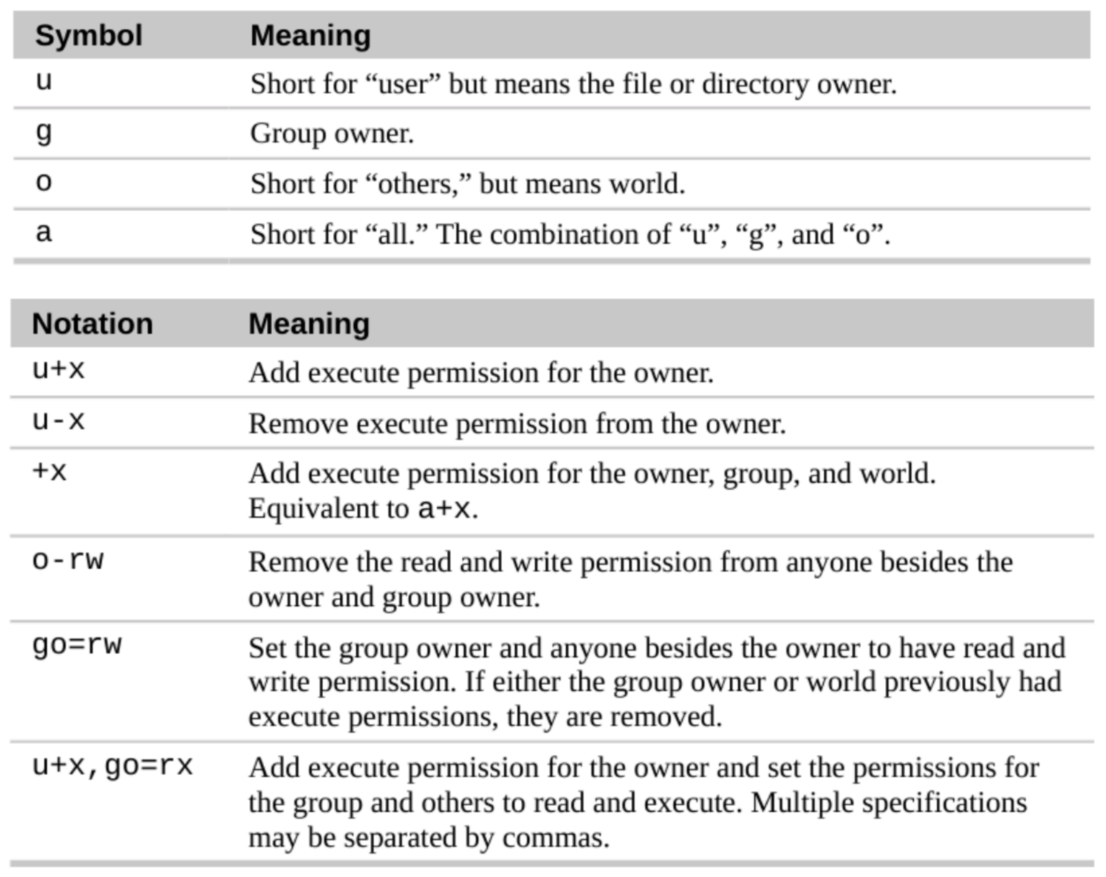
```
chown [option] new_owner file
chgrp [option] new_group file
```



```
$ ls -l
total 12
-rw-rw-r--  1 Rich sales 1521 Jan 19 15:38 customers.txt
-rw-r--r--  1 Christine sales  479 Jan 19 15:37 research.txt
-rw-r--r--  1 Christine sales  696 Jan 19 15:37 salesdata.txt
$ sudo chown Christine customers.txt
$ ls -l
total 12
-rw-rw-r-- 1 Christine sales 1521 Jan 19 15:38 customers.txt
-rw-r--r-- 1 Christine sales  479 Jan 19 15:37 research.txt
-rw-r--r-- 1 Christine sales  696 Jan 19 15:37 salesdata.txt
$ sudo chgrp marketing customers.txt
$ ls -l
total 12
-rw-rw-r-- 1 Christine marketing 1521 Jan 19 15:38 customers.txt
-rw-r--r-- 1 Christine sales      479 Jan 19 15:37 research.txt
-rw-r--r-- 1 Christine sales      696 Jan 19 15:37 salesdata.txt
```

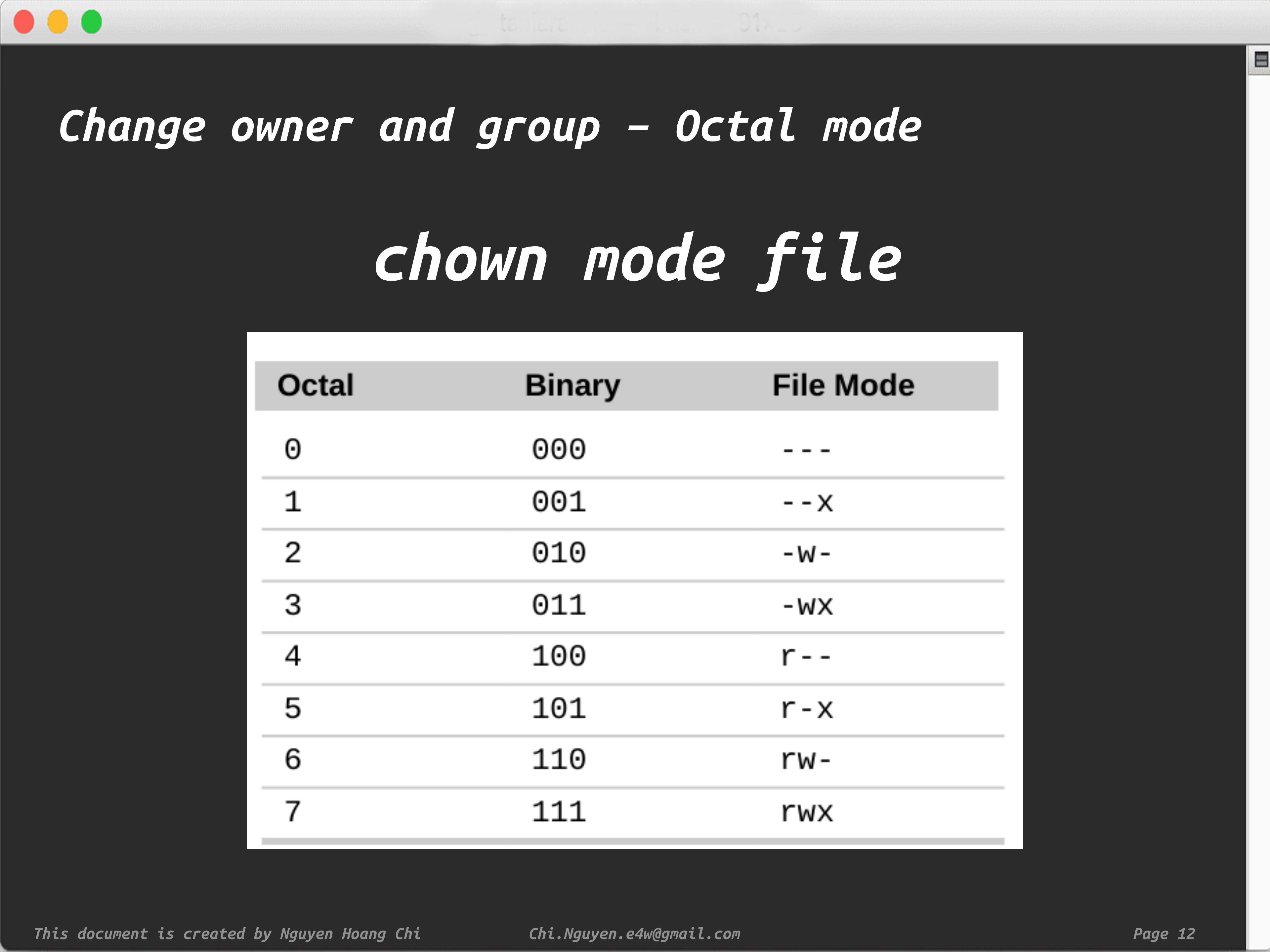
Change owner and group – Symbolic mode

chown mode file



Symbol	Meaning
u	Short for “user” but means the file or directory owner.
g	Group owner.
o	Short for “others,” but means world.
a	Short for “all.” The combination of “u”, “g”, and “o”.

Notation	Meaning
`u+x`	Add execute permission for the owner.
`u-x`	Remove execute permission from the owner.
`+x`	Add execute permission for the owner, group, and world. Equivalent to `a+x`.
`o-rw`	Remove the read and write permission from anyone besides the owner and group owner.
`go=rw`	Set the group owner and anyone besides the owner to have read and write permission. If either the group owner or world previously had execute permissions, they are removed.
`u+x,go=rx`	Add execute permission for the owner and set the permissions for the group and others to read and execute. Multiple specifications may be separated by commas.



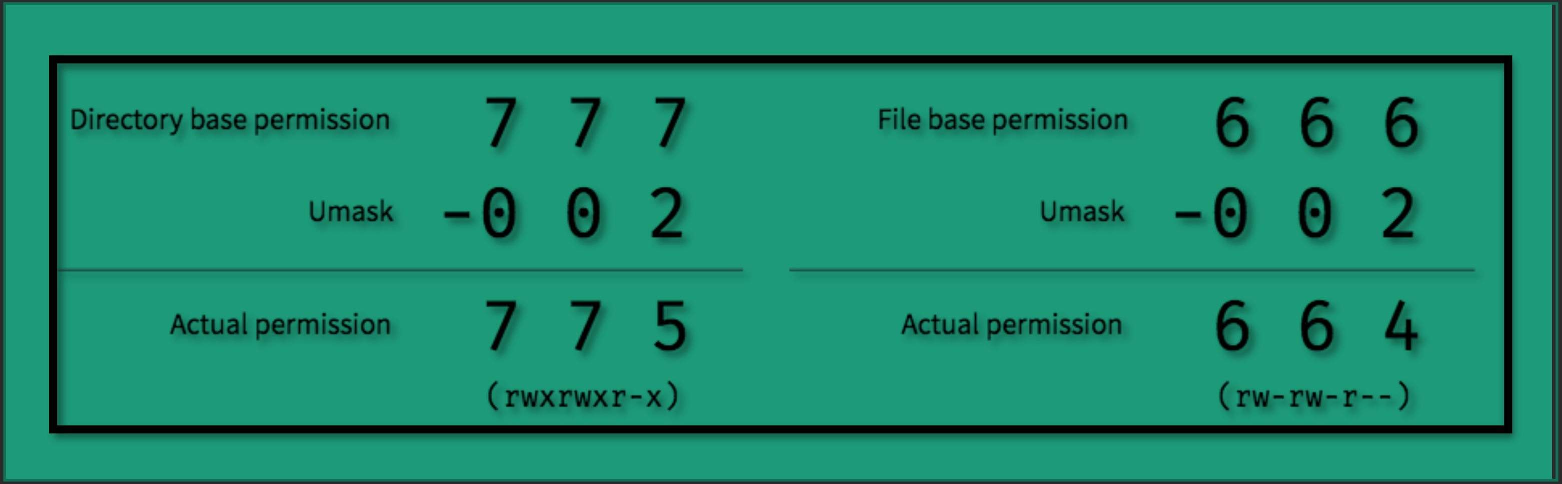
Change owner and group - Octal mode

```
chown mode file
```

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

Default permission

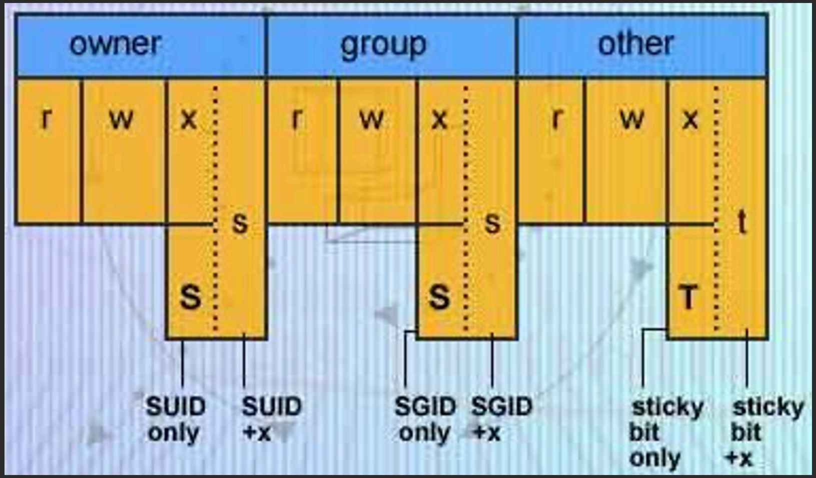
umask

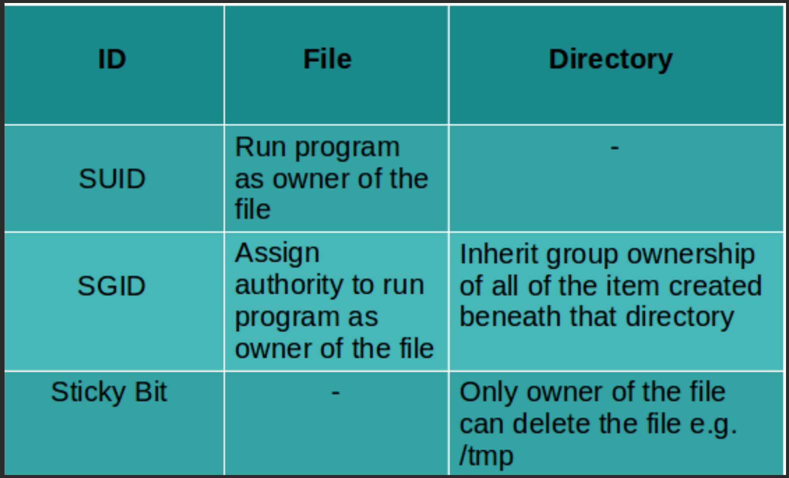


Directory base permission	7 7 7	File base permission	6 6 6
Umask	-0 0 2	Umask	-0 0 2
Actual permission	7 7 5 (rwxrwxr-x)	Actual permission	6 6 4 (rw-rw-r--)

Special access mode

SUID (Set user id), SGID (set group id), Sticky bit





ID	File	Directory
SUID	Run program as owner of the file	-
SGID	Assign authority to run program as owner of the file	Inherit group ownership of all of the item created beneath that directory
Sticky Bit	-	Only owner of the file can delete the file e.g. /tmp

Permission	Changing in Symbolic mode	Changing in Octal mode
SUID	chmod u+s file_name	chmod 4750 file_name
SGID	chmod g+s file_name	chmod 2750 file_name
Sticky bit	chmod o+t file_name	chmod 1750 file_name

Find system files and place files in correct location

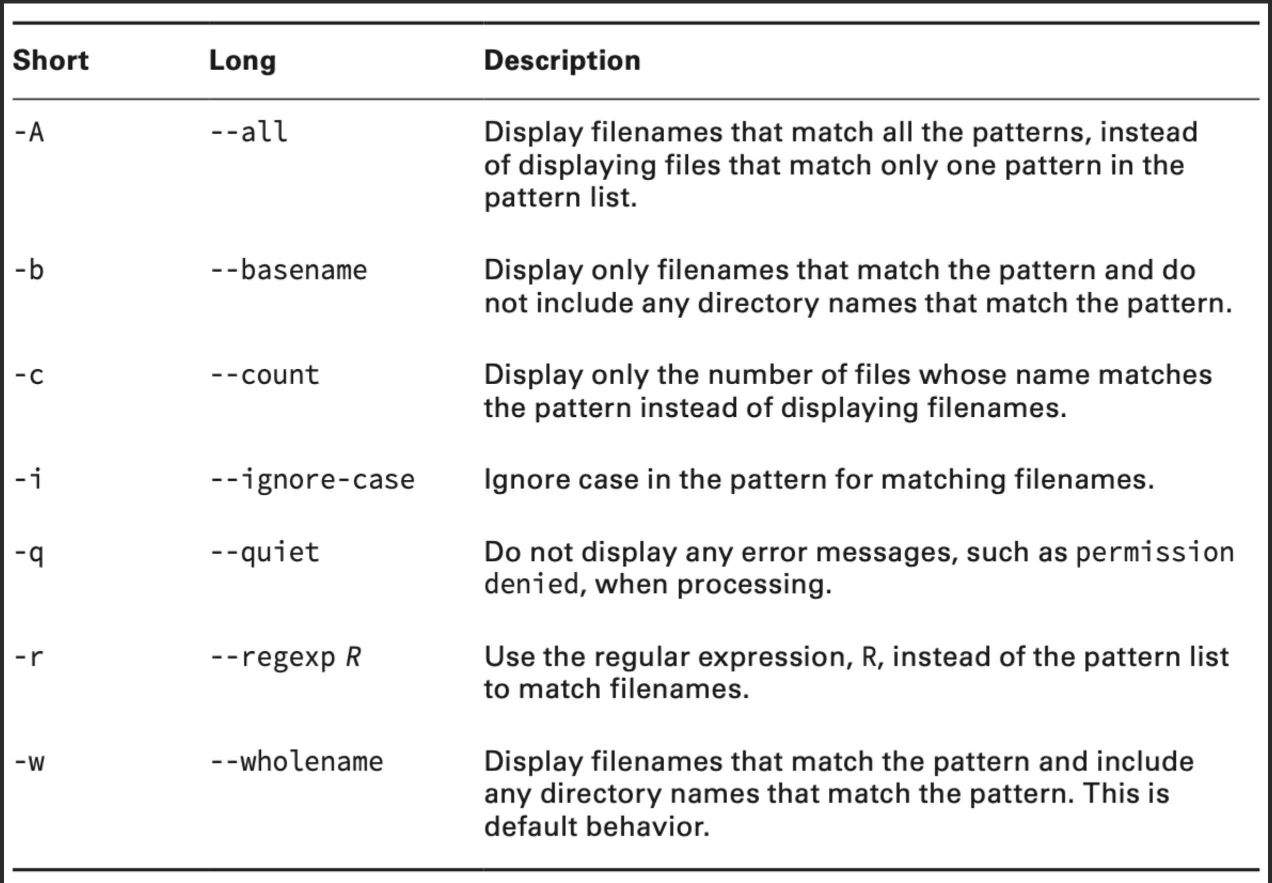
Locating files

which vs whereis

	which	whereis
Search path	$PATH	Linux standard paths; $PATH; $MANPATH
File type	Executable files	Executable files Source files Manual files

Locating files

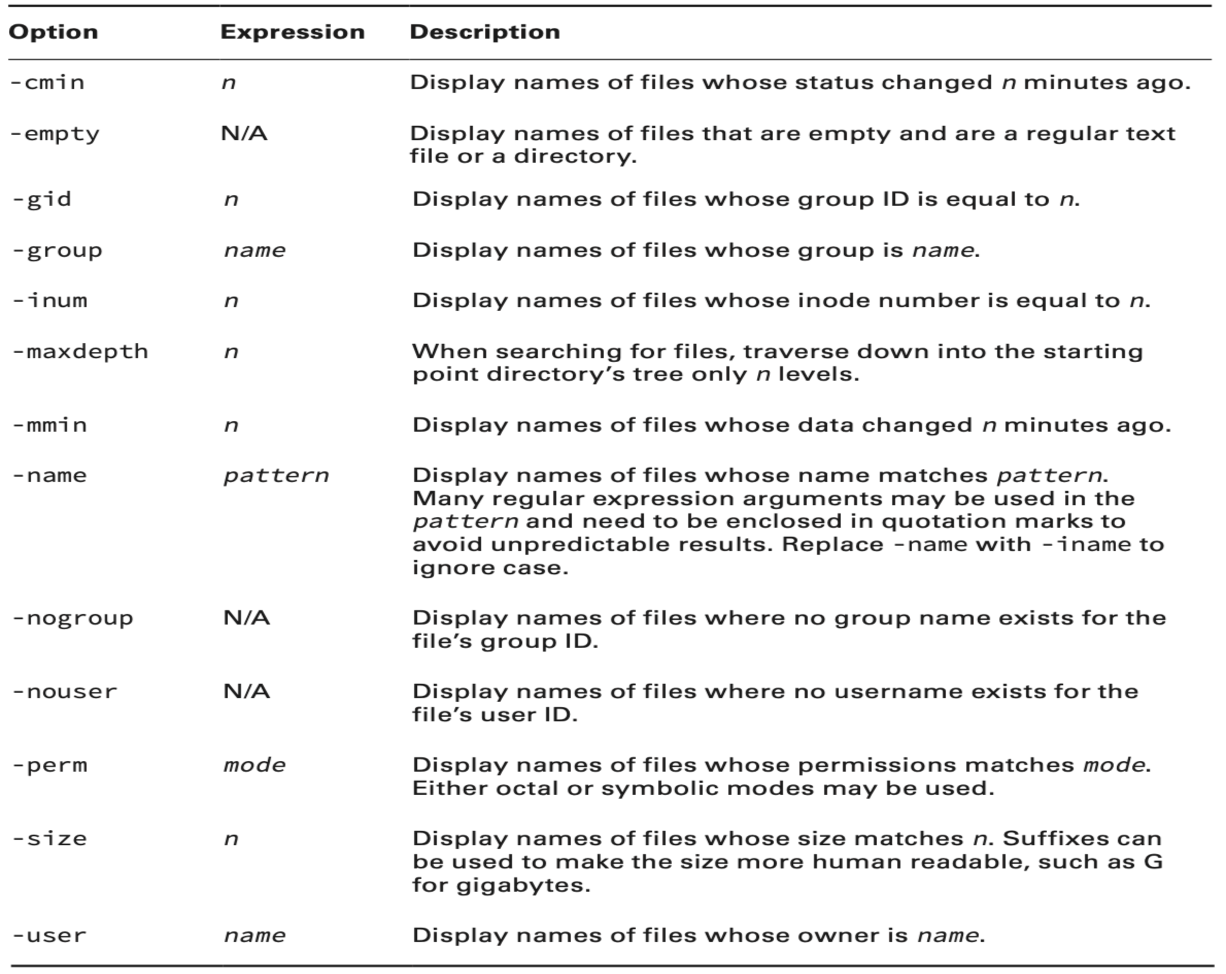
locate [option] pattern



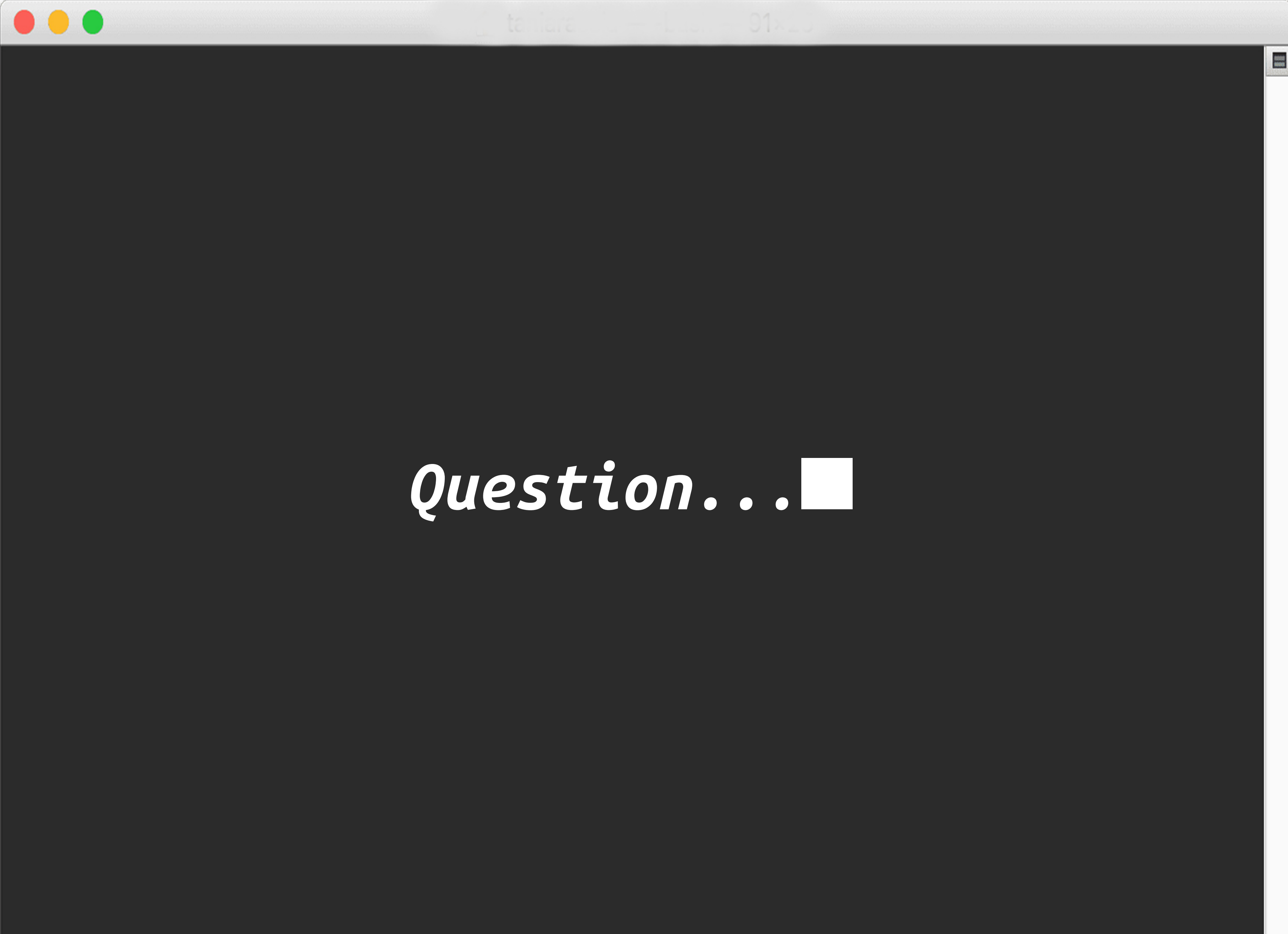
Short	Long	Description
`-A`	`--all`	Display filenames that match all the patterns, instead of displaying files that match only one pattern in the pattern list.
`-b`	`--basename`	Display only filenames that match the pattern and do not include any directory names that match the pattern.
`-c`	`--count`	Display only the number of files whose name matches the pattern instead of displaying filenames.
`-i`	`--ignore-case`	Ignore case in the pattern for matching filenames.
`-q`	`--quiet`	Do not display any error messages, such as `permission denied`, when processing.
`-r`	`--regexp` *R*	Use the regular expression, R, instead of the pattern list to match filenames.
`-w`	`--wholename`	Display filenames that match the pattern and include any directory names that match the pattern. This is default behavior.

Locating files

find [path] [option] pattern



Option	Expression	Description
-cmin	*n*	Display names of files whose status changed *n* minutes ago.
-empty	N/A	Display names of files that are empty and are a regular text file or a directory.
-gid	*n*	Display names of files whose group ID is equal to *n*.
-group	*name*	Display names of files whose group is *name*.
-inum	*n*	Display names of files whose inode number is equal to *n*.
-maxdepth	*n*	When searching for files, traverse down into the starting point directory's tree only *n* levels.
-mmin	*n*	Display names of files whose data changed *n* minutes ago.
-name	*pattern*	Display names of files whose name matches *pattern*. Many regular expression arguments may be used in the *pattern* and need to be enclosed in quotation marks to avoid unpredictable results. Replace -name with -iname to ignore case.
-nogroup	N/A	Display names of files where no group name exists for the file's group ID.
-nouser	N/A	Display names of files where no username exists for the file's user ID.
-perm	*mode*	Display names of files whose permissions matches *mode*. Either octal or symbolic modes may be used.
-size	*n*	Display names of files whose size matches *n*. Suffixes can be used to make the size more human readable, such as G for gigabytes.
-user	*name*	Display names of files whose owner is *name*.



Question...