CSCI 103: Introduction to Programming
Lab 12
Command Line and Relative Paths

Command Line

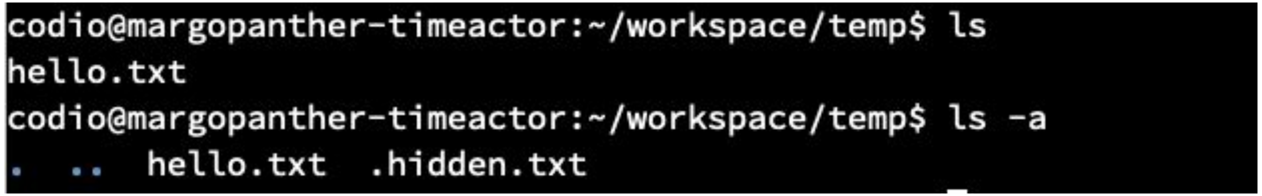
- Using the command line is an important skill to understand and navigate modern computer systems
- You can use the command line to more quickly and easily control file creation, location, and modification
- You will learn a few basic commands in this lab that you will then use to perform a kind of file-system "scavenger hunt". Have fun!

Commands To Use Today

- `ls` and `ls -l` and `ls -a`: **List** the files (and details) in the current directory. The flag "a" shows both visible and hidden files and directories.
- `cd folder-path`: **Change** directory to `folder-path` (`cd ~` brings you to the home directory)
- `cp src.cpp dest`: **Copy** the source file to the destination file or folder
- `mv old.cpp new.cpp`: **Move**/rename `old.cpp` to `new.cpp`
- `rm filename` or `rm -r folder`: **Remove** (delete) filename or folder (requires the -r flag for recursive removal)
- `mkdir folder`: **Make directory** (create a folder)
- `pwd`: Shows your present working directory. Helpful in remembering where you are in the file system.
- cat filename: Shows the content of the file

File Visibility

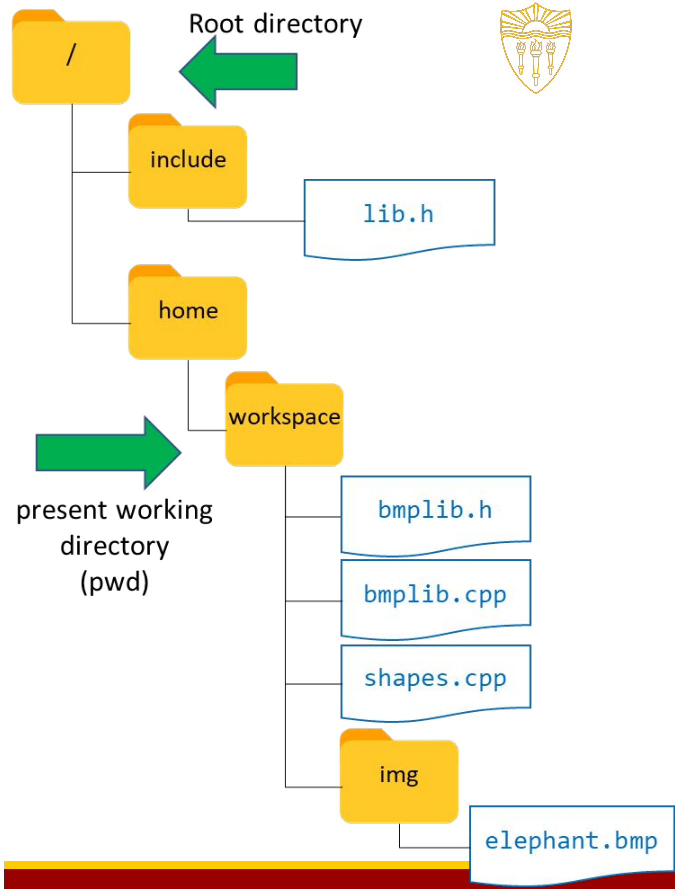
- Files and directories are hidden if their file name has a '.' as the first character of its name.
- Files and directories can be unhidden by renaming the file without the '.' at the front of the file name.



```
codio@margopanther-timeactor:~/workspace/temp$ ls
hello.txt
codio@margopanther-timeactor:~/workspace/temp$ ls -a
.   ..   hello.txt   .hidden.txt
```

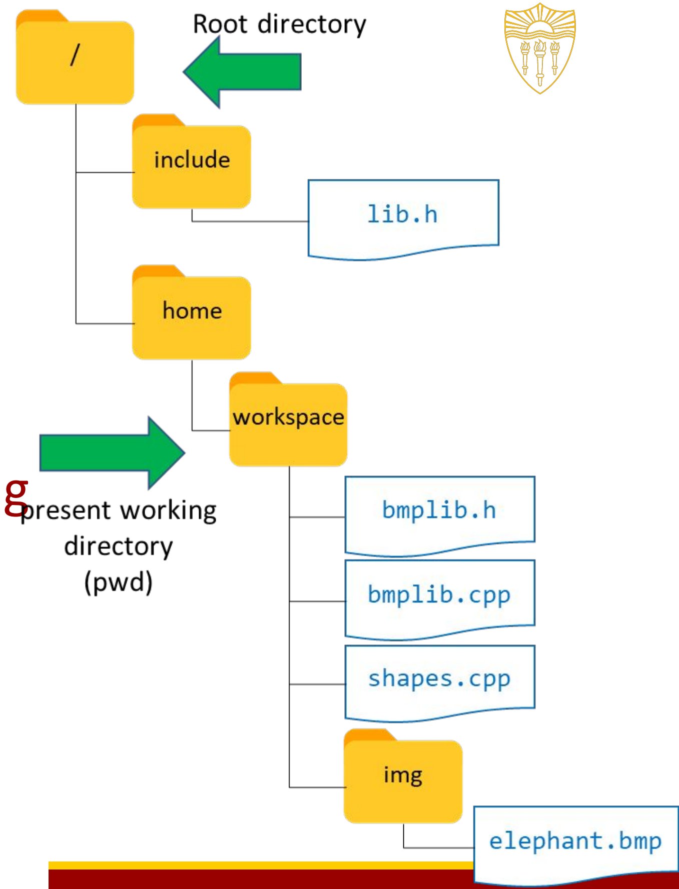
Absolute and Relative Paths

- An absolute path always starts its reference from the root folder (/)
  - Ex: /home/workspace/img/elephant.bmp
  - If the path starts with / the terminal interprets it as an absolute path
- A relative path implicitly starts from the present working directory
  - Ex: img/elephant.bmp
  - If the path does NOT start with / the terminal interprets it as a relative path
- You can use whatever path style is easier (usually relative paths save typing)



Path Shortcuts

- There are a few "alias" or shortcut paths
- A single dot (.) is a placeholder for the present working directory
  - `cp /home/workspace/img/elephant.bmp .`
    will copy elephant.bmp to the present working directory (i.e. /home/workspace)
- Two dots (..) is a placeholder for the parent folder (one level up from the present working directory)
  - `cd ..`
    Will change the present working directory to the parent folder



Sample Commands [1]

- Suppose our present working directory is `/home/workspace`
- We could
  - Change directory to img via

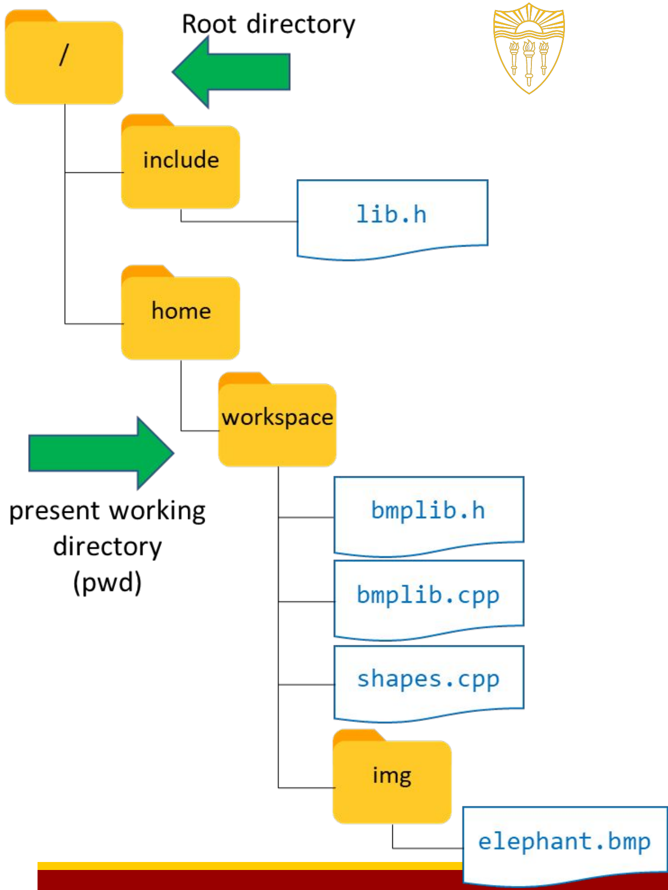
    ```
    $ cd img
    ```
  - Change directory to the root folder (/) via:

    ```
    $ cd ../..
    ```
    (relative path…OR…)

    ```
    $ cd /
    ```
    (absolute path)
  - List all the files in folder 2 levels up via:

    ```
    $ ls ../..
    ```
    Which might show:

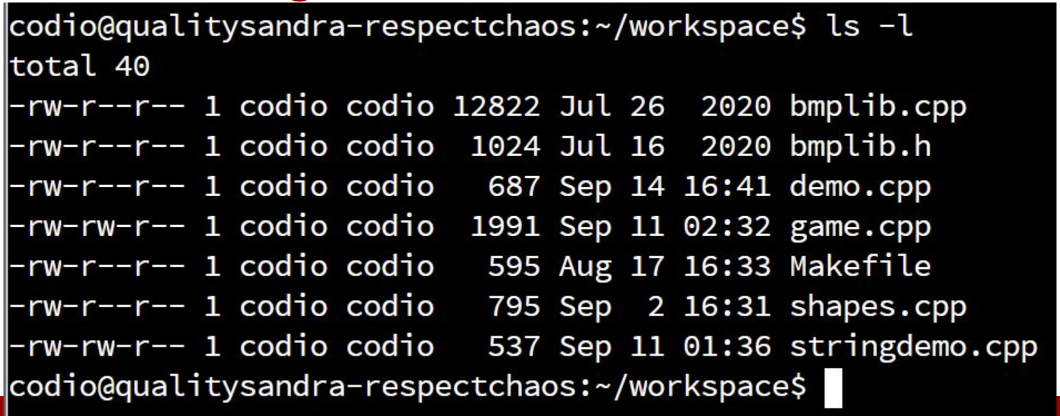
    ```
    include   home
    ```



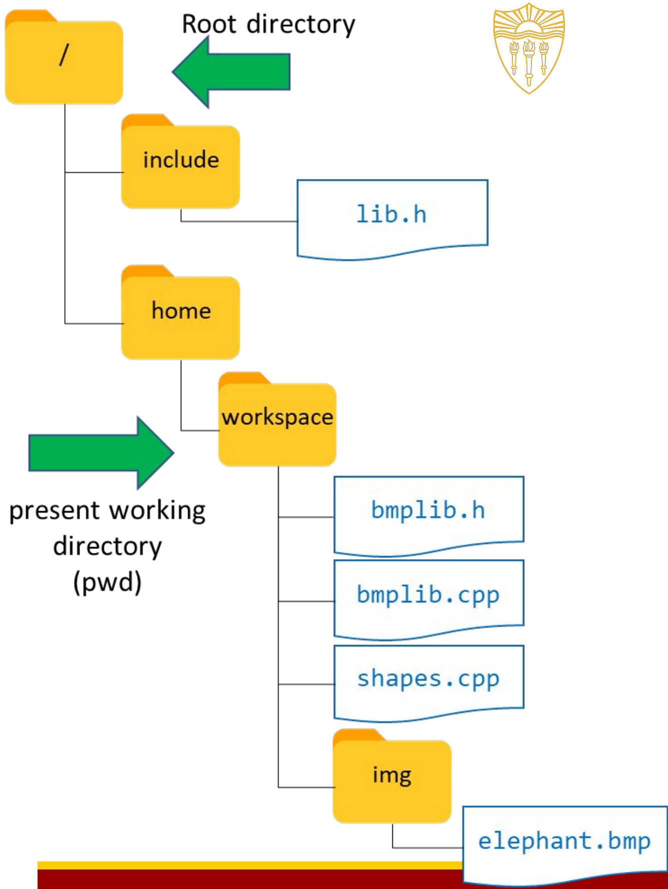
Sample Commands [2]

- Suppose our present working directory is /home/workspace
- We could
  - Print files/folders in the current folder with details
  - `$ ls -l`

    Which might show:



```
codio@qualitysandra-respectchaos:~/workspace$ ls -l
total 40
-rw-r--r-- 1 codio codio 12822 Jul 26  2020 bmplib.cpp
-rw-r--r-- 1 codio codio  1024 Jul 16  2020 bmplib.h
-rw-r--r-- 1 codio codio   687 Sep 14 16:41 demo.cpp
-rw-rw-r-- 1 codio codio  1991 Sep 11 02:32 game.cpp
-rw-r--r-- 1 codio codio   595 Aug 17 16:33 Makefile
-rw-r--r-- 1 codio codio   795 Sep  2 16:31 shapes.cpp
-rw-rw-r-- 1 codio codio   537 Sep 11 01:36 stringdemo.cpp
codio@qualitysandra-respectchaos:~/workspace$
```

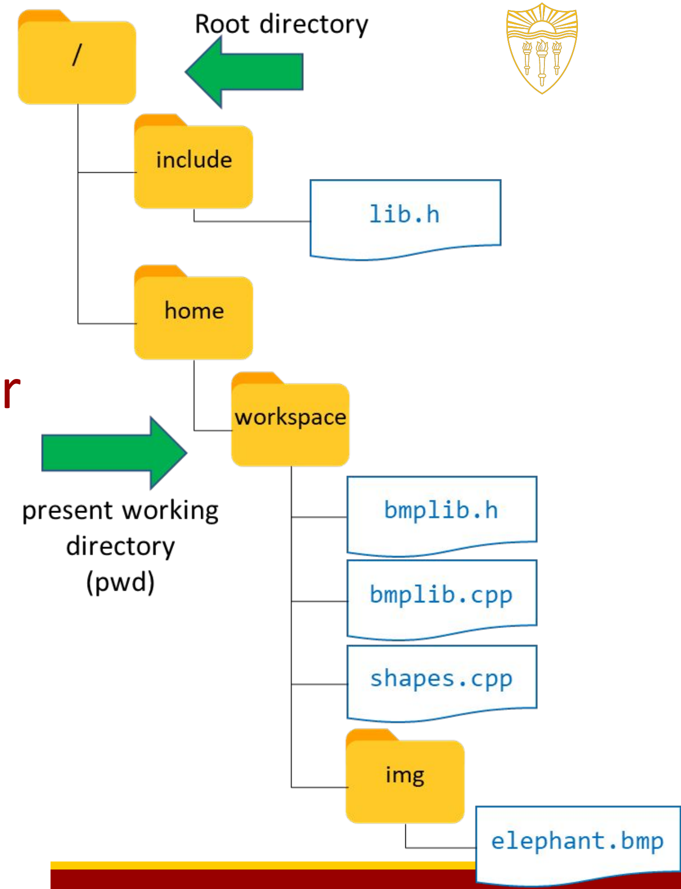


Sample Commands [3]

- Suppose our present working directory is `/home/workspace`
- We could
  - Copy lib.h from the /include folder to your current folder:
  - `$ cp ../../include/lib.h  .`
  - Move ALL files ending in .cpp to the img folder:

    `$ mv *.cpp img/`
  - Remove the elephant.bmp file

    `$ rm img/elephant.bmp`



Sample Commands [4]

- Suppose our present working directory is `/home/workspace`
- We could
  - Copy elephant.bmp to the root folder (using absolute paths)

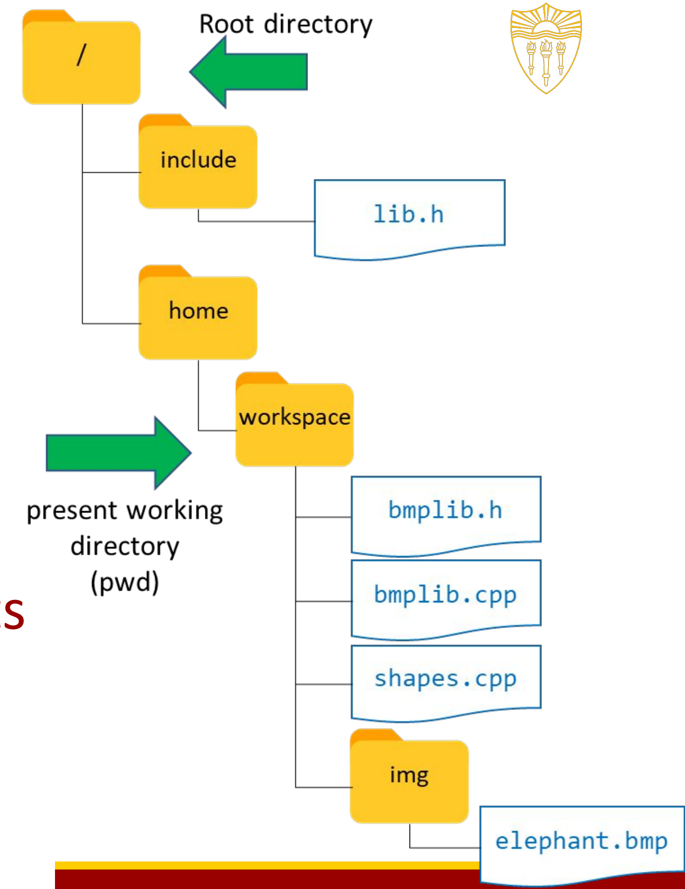
    ```
    $ cp img/elephant.bmp /
    ```

  - Remove the img <u>**folder**</u> and all its contents (including subfolders)

    ```
    $ rm -r img/
    ```

  - Rename shapes.cpp to app.cpp

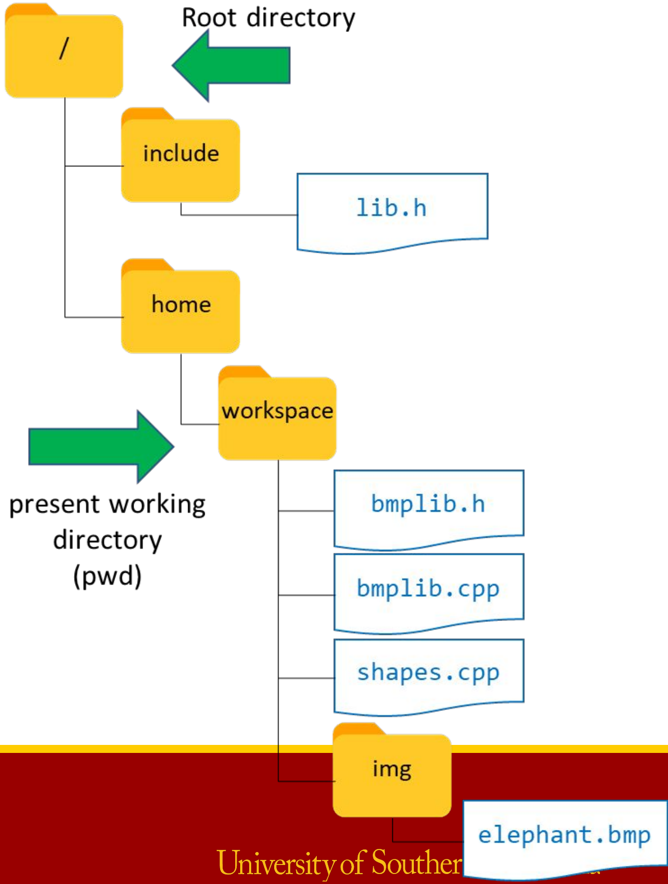
    ```
    $ mv shapes.cpp app.cpp
    ```



A Few More Commands

- cat and more: print a text file to the terminal screen for viewing (without an editor)

```
$ cat shapes.cpp
$ more ../../include lib.h
```



Command Reference

Command line utilities **ls**

$ ls [-flag_name]

- -a show all files and directories
- `-l` to see the file type, permissions, owner, file size and other information in detail
- -r display files in reverse order (original alphabetical order)
- -t list files in order of creation time
- Example: ls -l
    - to see details/modification dates of folders and files in the current directory

Command line utilities **mkdir**

```
$ mkdir [-p] dirName
```

- -p ensures that the directory exists, creates one if it does not exist
- Example: `mkdir dir_name`
  - In the current directory, create a subdirectory named dir_name
- Example: `mkdir -p dir_name/subdir_name`
  - In the dir_name directory under the current directory, create a subdirectory subdir_name
  - If the dir_name directory does not already exist, create one. (Note: If the -p parameter is not added in this example, and the original dir_name directory does not exist, an error will occur)

Relative Path vs Absolute Path

In simple words, an **absolute path** refers to the same location in a file system relative to the **root directory**, whereas a **relative path** points to a specific location in a file system relative to the **current directory** you are working on.

- Absolute Path Example:
  - /home/username/folder_name/filename.cpp
  - /home/username/folder_name/
- Relative Path Example:
  - ./filename.cpp - this represents the file in the current directory
  - ../folder/filename.cpp - move one level up and open the parent directory
  - ../../folder/filename.cpp - move two levels up and open the directory

Command line utilities **cd**

$ cd dirName

- dirName: The target directory to switch to
- Example: cd dir_name/subdir_name
  - Jump to dir_name/subdir_name
- Example: cd ../
  - Jump one level above the current directory

Command line utilities **rm**

$ rm [-r] filename

- filename: The file/folder to delete(remove)
- Example: rm test.txt
  - Delete file test.txt
- Example: rm -r homework
  - Delete folder homework and all containing files
- Example: rm -r *
  - Delete all files and folders in the current directory

Command line utilities **cp**

$ cp [-flag_name] source dest

- -r If the given source file is a directory file, all subdirectories and files in the directory will be copied
- -f Overwrite existing object files without prompting
- -i Give a prompt before overwriting the target file, asking the user to confirm whether to overwrite, and answering **y** the target file will be overwritten.
- Example: cp file.cpp dest_folder
  - Copy file.cpp to the directory dest_folder
- Example: cp -r files/ dest_folder
  - Copy all files in the directory files/ to the directory dest_folder

Command line utilities **mv**

$ mv source dest

- Move file/folder or rename file/folder
- Example: mv file.cpp new_file.cpp
  - Rename file.cpp to new_file.cpp
- Example: mv info/ logs
  - Put the info directory into the logs directory. Note that this command renames info to logs if the logs directory does not exist.
- Example: mv /usr/bin /*
  - Move all files and directories under /usr/bin to the current directory

Your Task

- You will now apply these commands to perform a kind of scavenger hunt which requires you to move, copy, and remove files
- You will run "tests" that ensure the desired file structure has been achieved
- Show your final results to the TAs to get credit and get your grade inputted
- Feel free to ask for help if you are stuck
- If you want to reset your lab for whatever reason, let a TA know