## Lab 0 Objectives

- Intro to Labs
- Intro to Operating Systems
- Start Lab 0
  - UNIX/Linux intro
  - Use emacs (Text Editor)
  - Register for Interactive Textbook
  - Canvas (Forum for “Broader Issues”)





## Intro to Labs

- Student Assistants
  - Jenna Bernstein ‘25
  - Fekry Mostafa ’25
- Tech Support Tom Marcais
  - Linux/CS account issues
  - tmarcais@wlu.edu

## Intro to Labs

- Typically: ~2 hours to get started on labs
  - ~1st hour is review (which is meant to help you get started on lab too)
  - Help from me and the student assistants
  - Today is not a typical lab!
- Often, will need to finish lab after lab period
  - Lab assignments are the majority of your homework
- Use this lab (P405), preferably, or P413
  - Or, work remotely *on these machines!*





## What Today Is and Is Not

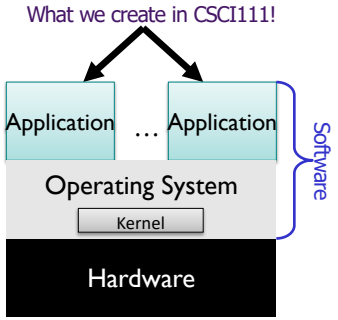
- Not ready for programming
- Set up for the rest of semester
- Develop skills
  - Communicating with computer
    - When we talk to computer, we need to be ***precise***
  - **Identifying problems** and **solving those problems**
- Start to learn Linux

# Basic Computer Architecture

Solve problems
MSOffice Applications (Excel, Word), Solitaire, Firefox, Internet Explorer

Manages hardware resources
Windows, OSX/11, UNIX, Android, Linux

The machine, made up of CPU, memory, hard drive, keyboard, etc.
Dell, Apple, HP, IBM, Toshiba, ...








# Parmly 405 Machines

- Run Linux, distribution: Ubuntu
- Parmly 413 is the "advanced lab" down the hall
  - Can use those machines when this lab is in use
- Use your W&L username and password to login
  - But, the **files** you have access to on the Linux machines are not the same as the files you have access to on other W&L lab machines

## Operating Systems

- Manage hardware resources
- Three popular desktop operating system variations:

| Mac | Windows | UNIX |
|---|---|---|

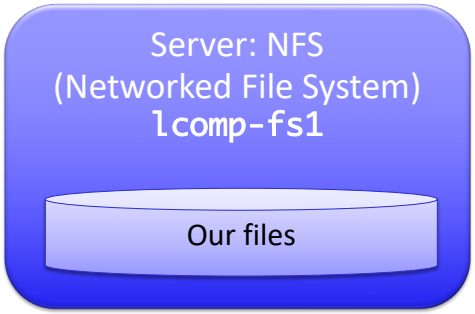
- Learn Linux (a UNIX variation) in this class

Macs are built on UNIX → can use UNIX commands



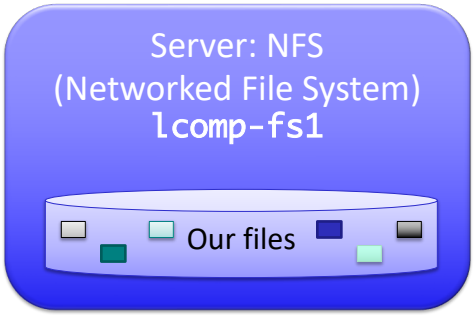


## CS Lab Architecture: File Server




- Stores files for the Computer Science department
  - Individuals' files, shared files

# CS Lab Architecture: File Server



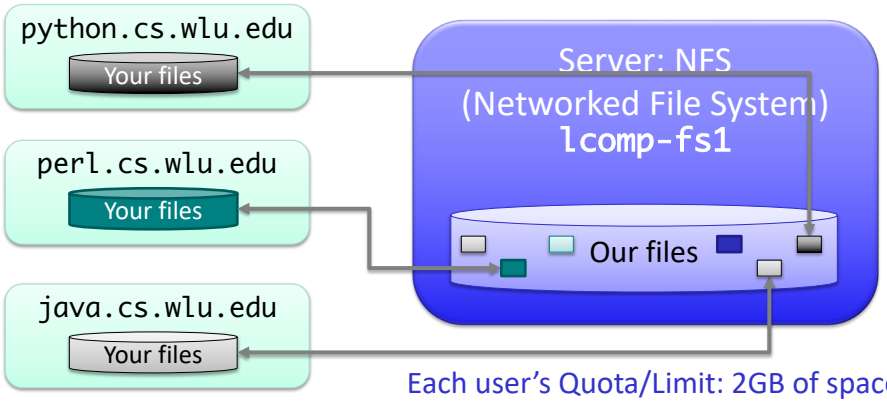

- Stores files for the Computer Science department
  - Individuals' files, shared files





# CS Lab Architecture

Lab machines




Each user's Quota/Limit: 2GB of space

## Getting Started

- You'll need a web browser

How can you launch a web browser?

  - Navigate to the course web site
- When you're done and leave lab, you should log out
  - **BUT** not shutdown the machine

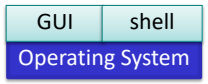
How do you log out?





## Intro to UNIX

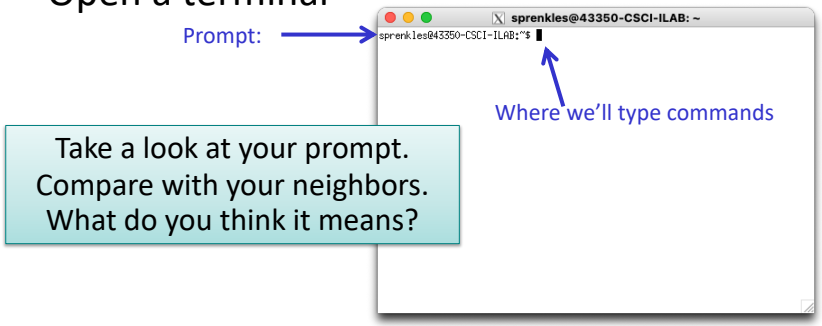
- Execute operations by typing commands in shell ***or*** using GUIs (Graphical User Interfaces)
- We will use both GUIs and command-line tools




- Pros and cons of command-line tools
  - Faster to use keyboard than mouse
  - Easier to repeat and automate
  - Larger learning curve, more error-prone, and can be intimidating

# Terminal

- Command-line interface to operating system
- Open a terminal




Take a look at your prompt.
Compare with your neighbors.
What do you think it means?





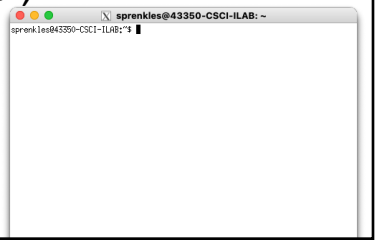
# Terminal

- Command-line interface to operating system
- Open a terminal

Prompt: `[username@machinename directoryIAmIn]$`

## UNIX Shortcuts: ~

- ~ represents ***your*** home directory
  - **Not** *the* home directory
  - Always with respect to the user
- When you open a new terminal, you're in *your* home directory








## GUI to Get Help

- At the prompt, type the command
  - `labhelp`
- Press enter

## Challenge: UNIX is a Bad Parent

- Doesn't tell you when you've done something right
- Only tells you when you've done something wrong

```
sprenkle@spartacus Desktop$ mv lab00.pptx.pdf lab00.pdf
sprenkle@spartacus Desktop$
```

Renames file from `lab00.pptx.pdf` to `lab00.pdf`

Since you didn't get an error message, it did something!
(May not be what you wanted/expected.)





## Intro to UNIX: Essential Commands

- Manipulating Files

| Command | What it does |
|---|---|
| `ls` | **ls**t the files, directories in a directory |
| `mkdir dname` | **m**a**k**e a **dir**ectory with the name *dname* |
| `cp src dest` | **cp**y a *src* to a *dest*<br>*src* can be a file, set of files, or a directory<br>*dest can be a file or a directory* |
| `rm file` | **rm**ove (delete) a file/directory |

- Navigating Directories

| | |
|---|---|
| `pwd` | **p**rint **w**orking **d**irectory |
| `cd name` | **c**hange to **d**irectory name |

## Command-Line Practice

- In the terminal, execute the following commands:
  - `pwd`
  - `ls`
  - `ls Documents`





## Home Directories

- When you entered the command pwd, the response was the path to your home directory:
  - /home/**username**@ad.wlu.edu, where *username* is your username
    - I will often shorthand this to just `/home/username`

## Useful Trick: Up Arrow

- Hit the up arrow.  What happened?

- Hit the up arrow again?  What happened?





## What is the Unix command to do the following?

In your rows, determine these commands

1. Find out what directory you're in
2. View the contents of the directory
3. Create a directory called cs111
4. View the contents of your directory (again)
5. Go into the cs111 directory
6. Find out what directory you're in
7. View the contents of cs111 directory

## What is the Unix command to do the following?

Now, execute those commands!

1. Find out what directory you're in
   - `pwd`

   You should be in your home directory
2. View the contents of the directory
   - `ls`

   What files are in your home directory?
3. Create a directory called cs111
   - `mkdir cs111`
4. View the contents of your directory again

   What files are in your home directory now?
   - `ls`
5. Go into the cs111 directory
   - `cd cs111`
6. Find out what directory you're in
   - `pwd`

   You should be in /home/username@ad.wlu.edu/cs111
7. View the contents of cs111 directory
   - `ls`





## Process for Determining a Command?

- Reflect on what your process was for determining a command
  - Giving a process makes it a little less daunting to do the task again

## Process for Determining Command

1. Figure out what you're trying to do → what command does that?
2. What additional information does that command need?



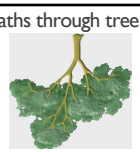


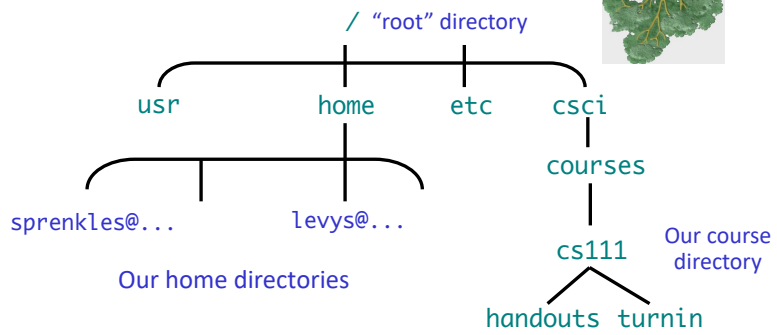
## Intro to UNIX: File Structure

- Organize our files
- Hierarchy of *directories* or "folders"

- Similar to what you have on your computer

# (Partial) Linux File Structure

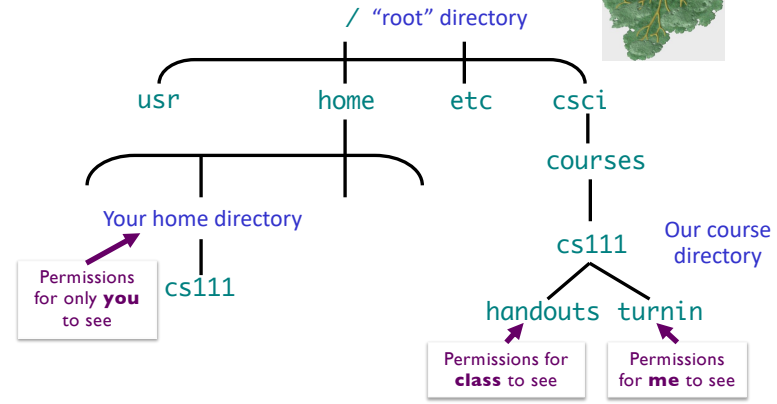




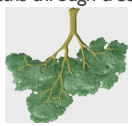


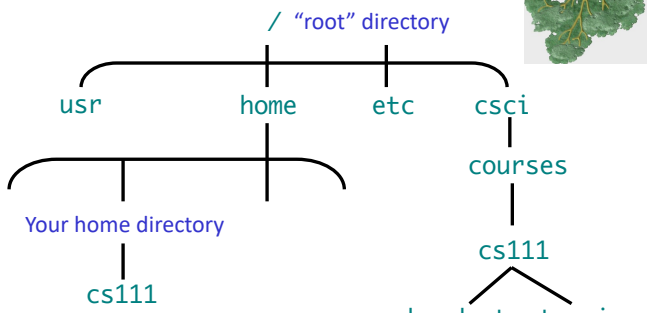

# (Partial) Linux File Structure

# (Partial) Linux File Structure

Paths through tree





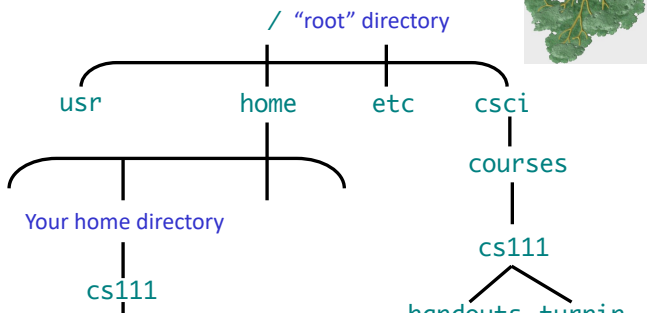

Consider if you are in your cs111 directory and you run `mkdir lab0`, how does the file structure change?





# (Partial) Linux File Structure

Paths through tree




You now have this directory!

Do this in the terminal!

## Paths

- ***Paths*** specify locations of files, directories
  - Used in a variety of commands
- Two types of paths: absolute and relative





## Absolute vs Relative Paths

Paths through tree

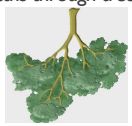


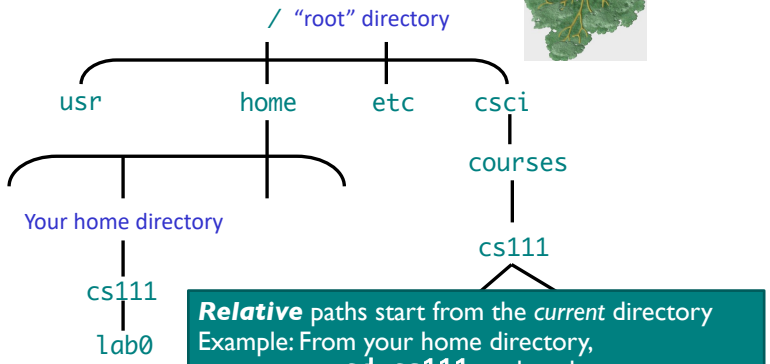

***Absolute*** paths start from the root directory (/), work no matter where you are
→ What you see when you type pwd

What is the absolute path to the cs111 directory?

# Absolute vs Relative Paths

Paths through tree






***Relative*** paths start from the *current* directory
Example: From your home directory,
you can type **`cd cs111`** rather than
`cd /home/username@ad.wlu.edu/cs111`





# Absolute vs Relative Paths

**Absolute**

- Always start at the root: /
- Absolute is kinda like always giving directions from Elrod Commons, where Elrod is our root
  - Benefit: directions always work!
- Made up example to get to Parmly 405
  - /Leyburn/ScienceCenter/Parmly/405

/ = Elrod Commons

**Relative**

- Start from current directory
- Relative is giving direction from where you are
  - **But**, only work if you're in that location
- Made up example to get to Parmly 405, where you're in the Science Center
  - Parmly/405
  - This path won't work if you're in the CGL

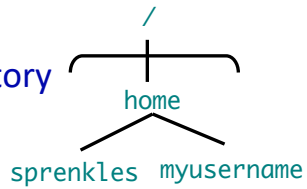
Takeaway: Either can be used to specify a path. With experience, you'll know which to use when.

# Intro to UNIX: Shortcuts

| Shortcut | Meaning |
|---|---|
| . | Current Directory |
| .. | Parent Directory |

- Often used with `cp`, `mv`, `cd` commands

- **`cd`** or **`cd ~`**
  - Change to *your* HOME directory



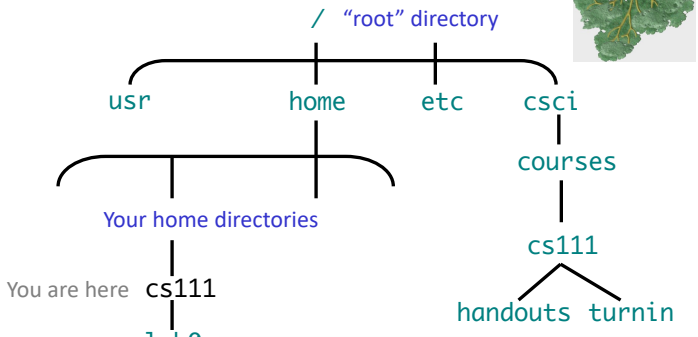

Example: `/home` is the ***parent directory*** of `/home/sprenkles@ad.wlu.edu`





# Absolute vs Relative Paths

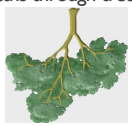
Paths through tree

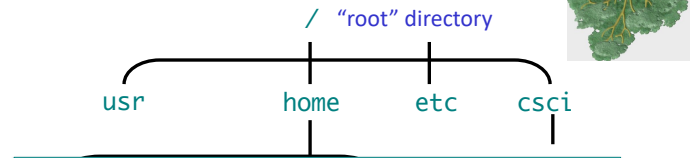



Given that you're in *your* cs111 directory, how would you get to *lab0*? To your *home* directory?
To the *handouts* directory?
- Provide 1) the *absolute* path and 2) the *relative* path

# Absolute vs Relative Paths

Paths through tree



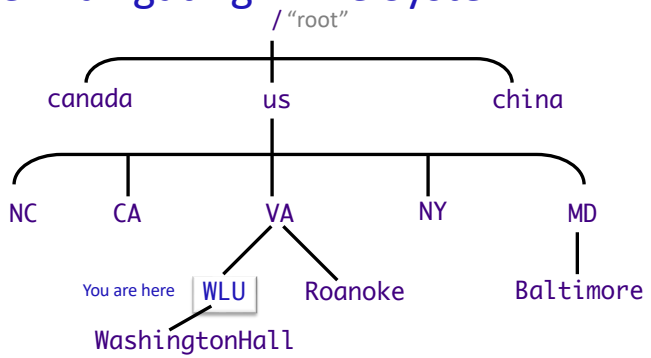



- To lab0
  - cd /home/username/cs111/lab0
  - cd lab0
- To home
  - cd /home/username@ad.wlu.edu
  - cd ..
  - cd
  - cd ~
- To handouts
  - cd /csci/courses/cs111/handouts
  - cd ../../../csci/courses/cs111/handouts





# Practice: Navigating A File System




- Given that you're at **WLU**, how would you get to Washington Hall? To Roanoke? To Baltimore?
  - Use either absolute or relative path, whichever is easier

# Practice, with Tab Completion

*This is an absolute path*

- Goal: go to the directory `/csci/courses/2022_23_winter-csci_111_01`
  - You can use ***tab completion*** to help you complete commands
  - After typing the appropriate command, start to type `/cs` and then press tab.
    - What happens?
  - Now that you're in the `/csci` directory, press tab twice
    - What do you see?
  - Use tab completion to help you complete the rest of the path
- Aside: to make navigation easier, I have linked the name `cs111` to that long name (`2022_23_winter-csci_111_01`) so we don't need to use the long name
- What are the contents of this directory?
- How can you get to the directory `/csci/courses`?
- How can you get back to *your* home directory? (3 ways)





# Opening a Text Editor

- Text editor: an application to write/edit text files
  - Text files: program source code, HTML code
  - Like NotePad++ or TextEdit
- To run one text editor:
  - `emacs &`
  - `&` means "run in the background" so you can keep using the terminal

## emacs: A text editor

Check: are you are in your home directory? Now, go into your `cs111` directory.

- `emacs &`
  - Command to run
- Create a new file (under File → Visit New File), add some text to it
  - e.g., "this is my file"
- Save the file in *your* `cs111` directory, naming it `test.txt`
- Exit emacs from the menu
- What are the contents of the directory now?





## More on the `cp` command

- `cp src dest`
  - `src`: what you want to copy
  - `dest`: to where you want to copy
    - If `dest` is a directory, copies `src` into that directory
    - If `dest` is not a directory, makes a copy of `src` and names it `dest`
- Practice in the terminal:
  - If needed, go to the location of test.txt
  - Copy the file you just created to make a backup of it, e.g., named `test.txt.bkup`
  - Check that that worked
  - Copy the original (test.txt) to your lab0 directory
  - Check that that worked

First, discuss the steps

## Using the Wildcard: *

- Go into `/csci/courses/cs111/handouts/lab0`
  - What are the contents of this directory?
- Try executing
  - `ls *.py`
  - `ls example.*`

What does the * do?





## Wildcard: *

- Match 0 or more characters in filenames
- Used to operate on more than one file
- Follow up question: What does * on its own do?

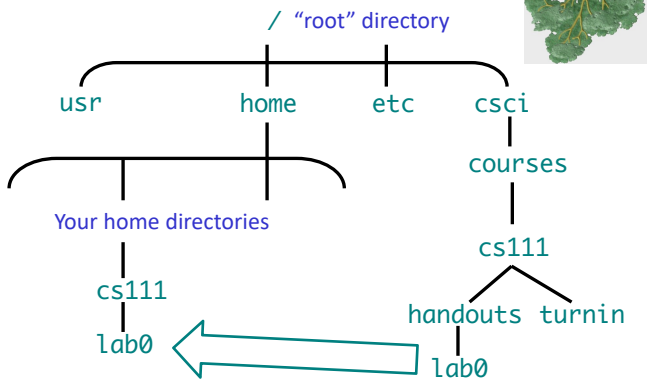
## Reset!

- Go to your home directory!
- Now go into your cs111/lab0 directory





## New Copy Task


**Task: Copy all the files from the course's lab0 to my lab0 directory**

## Breaking Down the Task

- What do we want to do?
- What command should we use?
- What does that command require?





## Breaking Down the Task

- What do we want to do? Copy!
- What command should we use? cp
- What does cp require? A source and destination
  - What is the source?
  - What is the destination?
  - How should we specify those directories?
    - Keep in mind: Where are we?

## Breaking Down Task

- What do we want to do? Copy!
- What command should we use? cp
- What does cp require? A source and destination
  - What is the source? The files in the course's handouts/lab0 directory
  - What is the destination? My cs111/lab0 directory
- How should we specify those directories?
  - Keep in mind: Where are you? I am in my cs111/lab0 directory
- Bring it all together:

```
cp /csci/courses/cs111/handouts/lab0/* .
```





## Linux Quiz

- True or False: I should shut down the machine when I am done using it.
- True or False: My CS account is the same as my W&L account.
- True or False: I can give my password to my friend who needs to access my account.

## Creating a Web Page

- Practical application of UNIX command skills
  - Practice commands you learned today
- Learning from following examples and adapting
- Learn what's "behind the curtain" of web pages





## CS Department's Web Server


- Holds files that we want to expose to the world through the web
- Separate file system from the CS file system
- Requires special permissions to be able to access
  - YOU have that special permission!

## ssh: secure shell


- Allows you to remotely log into the web server
  - Create web pages
- Just like if we were directly on the machine





## ssh: secure shell


- Allows us to remotely log into a lab machine!
- Use just like if we were directly on the machine

## Whew!

- You hung in there!
- You learned a lot! (I hope!)
- You didn't back down!
- But…. We're not done yet





## Lab 0 Checklist

- ✓ Linux
- Go to Browser, Lab 0 Page
  - Remote access to the lab machines
  - Canvas discussion forum
  - Interactive textbook
  - Canvas: introductory survey

Due Friday before class