

Objectives

- Review Linux
- Introduction to
 - Problem solving
 - Programming languages

Review: Lab

- Learned some UNIX commands
- Created a Web page

What did you learn?

Review: Lab

- Learned some UNIX commands
- Created a Web page
- Lessons learned:
 - Problems are fixable (often just typos!)
 - No “sorry” → you’re learning
 - Learn from, adapt examples
 - Find a good solution

Review: UNIX

- UNIX is a bad parent
 - Doesn’t tell you when you’ve done something right
 - Only tells you when you’ve done something wrong

Terminal:

```
sprenkle@spartacus Desktop$ cp lab00.ppt.pdf lab00.pdf
sprenkle@spartacus Desktop$
```

CORRECT! Because didn’t get an error message!

Review: Linux

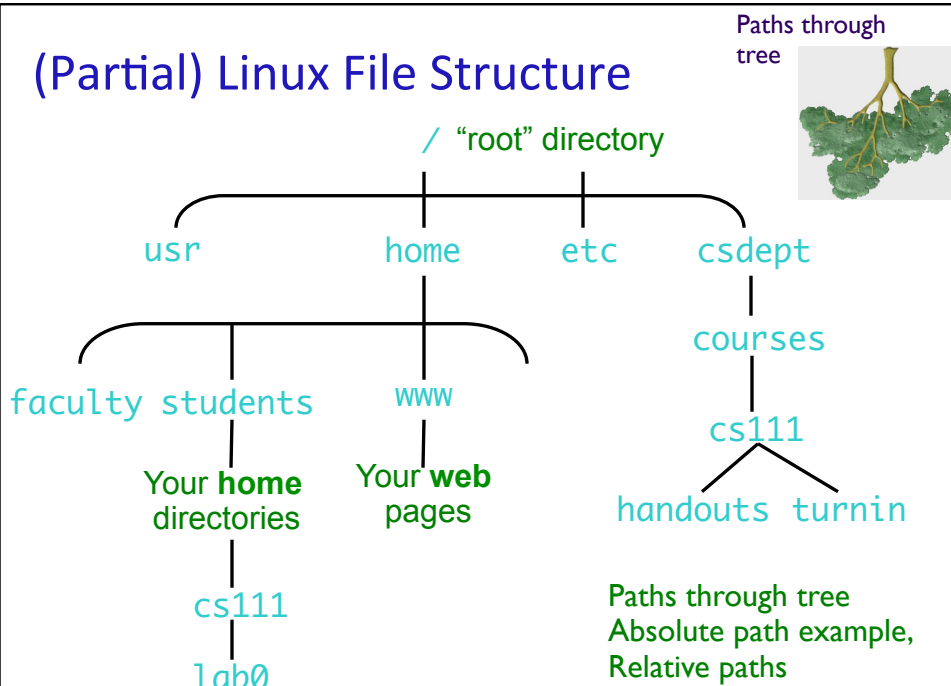
- How do you ...
 - List the files in a directory?
 - Change your current directory?
 - Make a directory?
 - Find out the current directory?
 - Make copies of files?
- What is the shortcut for ...
 - The current directory?
 - The parent directory?
 - Your home directory?

Jan 10, 2018

Sprenkle - CSCI111

5

(Partial) Linux File Structure

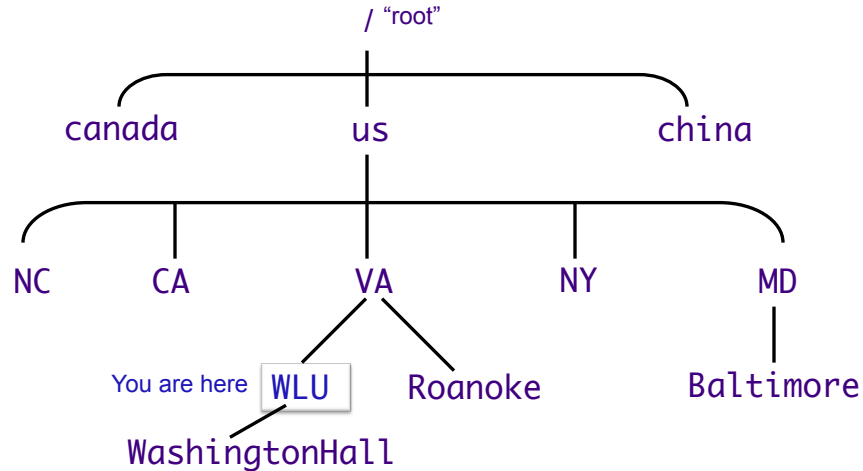


Jan 10, 2018

Sprenkle - CSCI111

6

Relative Paths vs Absolute Paths



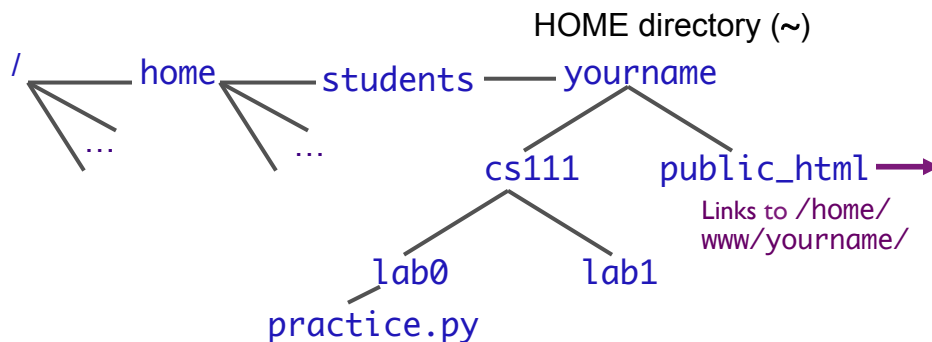
- Given that you're at **WLU**, how would you get to Washington Hall? To Roanoke? To Baltimore?

Jan 10, 2018

Sprenkle - CSCI111

7

Review: Linux File System



~ is a shortname for your home directory, i.e., short for /home/students/yourname

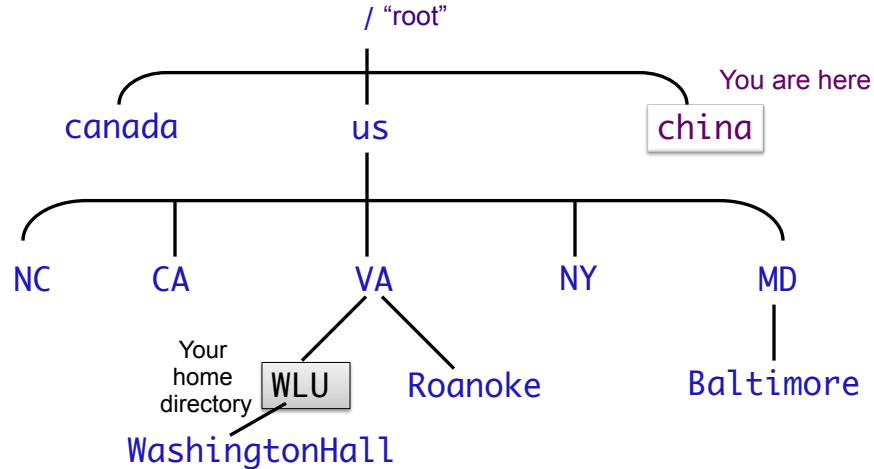
- What is the *syntax* for the copy command?
- How would you copy `practice.py` to your `public_html` directory if you were in `public_html`? If you were in `cs111`?

Jan 10, 2018

Sprenkle - CSCI111

8

Relative Paths vs Absolute Paths



- Given that you're in **China**, how would you go to Canada? WLU? Washington Hall?

Jan 10, 2018

Sprenkle - CSCI111

9

Review: Labs

- "That's it?"
 - Often, students get overwhelmed by the directions, but then the work isn't that difficult
- Worth 34% of your grade
 - Should get in B+/A- range *easily* with help from student assistants and me

Jan 10, 2018

Sprenkle - CSCI111

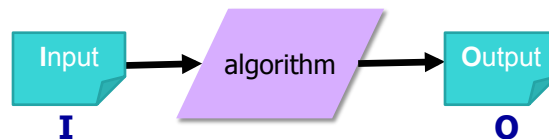
10

Review:

Computational Problem Solving 101

- Algorithm: a well-defined recipe for solving a problem
 - Has a finite number of steps
 - Completes in a finite amount of time
- Program
 - An algorithm written in a programming language
 - Also called code
 - Large programs, solving many problems together → application

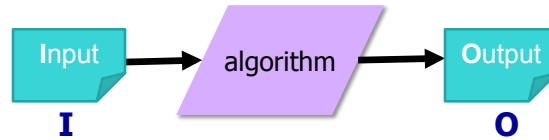
Algorithms: Input and Output



- Algorithms often have a defined **input** and **output**
- **Correct** algorithms give the intended output for a set of input
- Example: Multiply by 10
 - I/O for a correct algorithm:
- More examples
 - averaging numbers, recipes

Input	Output
5	50
.32	
x	

Algorithms: Input and Output



- Algorithms often have a defined **input** and **output**
- **Correct** algorithms give the intended output for a set of input
- Example: Multiply by 10
 - I/O for a correct algorithm:
- More examples
 - averaging numbers, recipes

Input	Output
5	50
.32	3.2
x	10x

Making a Peanut Butter & Jelly Sandwich

- How do you make a peanut butter and jelly sandwich?
- Write down the steps so that someone else can follow your instructions
 - Make no assumptions about the person's knowledge of PB&J sandwiches
 - The person has the following materials:
 - Loaf of bread, Jar of PB, Jar of Jelly
 - 2 knives, paper plates, napkins

Discussion of PB&J

- The computer: a blessing and a curse
 - Recognize and meet the challenge!
- Be unambiguous, descriptive
 - Must be clear for the computer to understand
 - “Do what I **meant!** Not what I said!”
 - Motivates programming languages
- Creating/Implementing an algorithm
 - Break down pieces
 - Try it out
 - Revise

Discussion of PB&J

- Steps need to be done in a particular order
- Be prepared for special cases
 - Any other special cases we didn't discuss?
- Aren't necessarily spares in real life
 - Need to write correct algorithms!
- Reusing similar techniques
 - Do the same thing with a little twist
- Looping
 - For repeating the same action

Parts of an Algorithm

- Input, Output
- Primitive operations
 - What data you have, what you can do to the data
- Naming
 - Identify things we're using
- Sequence of operations
- Conditionals
 - Handle special cases
- Repetition/Loops
- Subroutines
 - Call, reuse similar techniques

An overview for the semester!

TODO

- Lab 0 due on Friday
- Broader Issue Writeup Due on Sakai by 11 a.m. Friday
 - <http://www.cs.wlu.edu/~sprenkle/cs111/csissues.php#writeup>