

Objectives

- 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

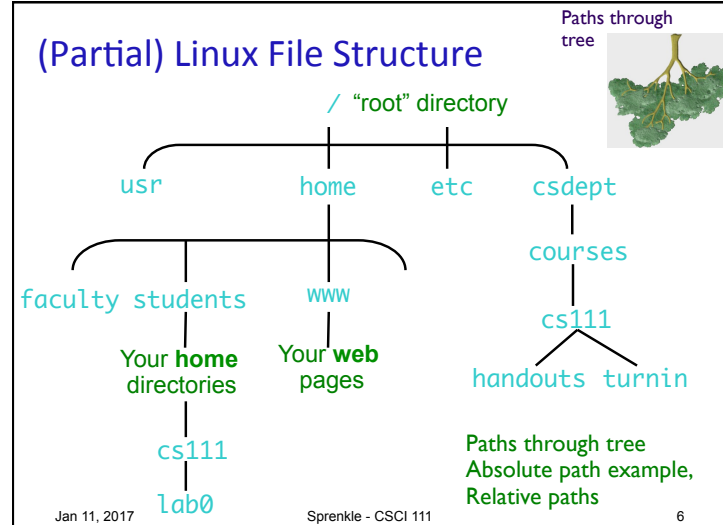
- How do you ...
 - List the files in a directory?
 - Change your current directory?
 - Make a directory?
 - Find out the current directory?
 - Make copies of files?
- What is the shortcut for ...
 - The current directory?
 - The parent directory?
 - Your home directory?

Jan 11, 2017

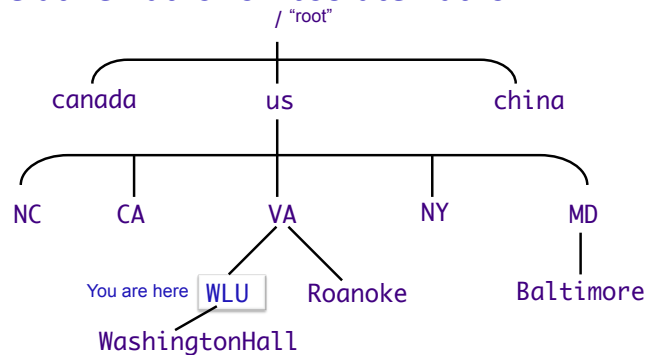
Sprengle - CSCI 111

5

(Partial) Linux File Structure



Relative Paths vs Absolute Paths



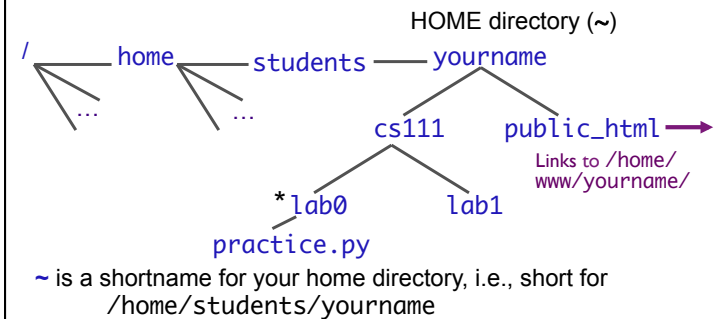
- Given that you're at **WLU**, how would you get to Washington Hall? To Roanoke? To Baltimore?

Jan 11, 2017

Sprengle - CSCI 111

7

Review: Linux File System



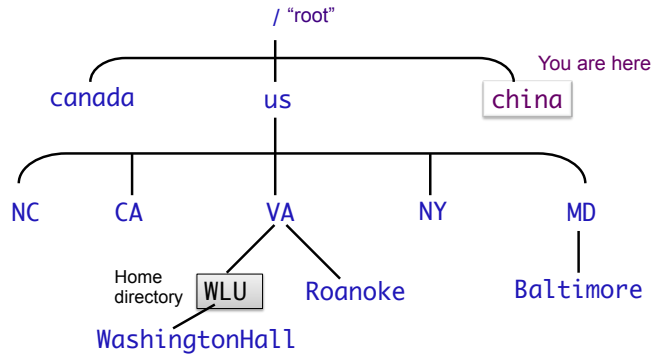
- What is the *syntax* for the copy command?
- How would you copy `practice.py` to your `public_html` directory if you were in `public_html`? If you were in `labs`?

Jan 11, 2017

Sprengle - CSCI 111

8

Relative Paths vs Absolute Paths



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

Jan 11, 2017

Sprengle - CSCI 111

9

Review: Labs

- "That's it?"
 - Often, students get overwhelmed by the directions, but then the work isn't that difficult
- Worth 38% of your grade
 - Should get in B+/A- range *easily* with help from student assistants and me

Jan 11, 2017

Sprengle - CSCI 111

10

Computational Problem Solving 101

- Computational Problem:**
A problem that can be solved by logic
- To solve the problem:
 - Create a **model** of the problem
 - Design an **algorithm** for solving the problem using the model
 - Write a **program** that *implements* the algorithm

Jan 11, 2017

Sprengle - CSCI 111

11

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

Jan 11, 2017

Sprengle - CSCI 111

12

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

Jan 11, 2017

Sprenkle - CSCI 111

13

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

Jan 11, 2017

Sprenkle - CSCI 111

14

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

Jan 11, 2017

Sprenkle - CSCI 111

15

Discussion of PB&J

- Steps need to be done in 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

Jan 11, 2017

Sprenkle - CSCI 111

16

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!

Other Lessons To Remember

- A cowboy's wisdom: Good judgment comes from experience
 - How can you get experience?
 - Bad judgment works every time
- Program errors can have **bad** effects
 - Prevent the bad effects--especially before you turn in your assignment!

"Really?" with Professor Sprenkle

- In *TV Guide*, showrunners of *Once Upon a Time* were asked, "Give us an algorithm for your show."
 - Example: 1 part *Snow White* + 1 part *Lost* + .5 *Alias*
- They said, "We don't understand math. That's why we became writers."

Looking Ahead

- Friday: Broader Issue Discussion
- Programming!