

Lab 0 Objectives

- Intro to Labs
- Intro to Operating Systems
- Why programming languages?
- Start Lab #0
 - UNIX/Linux intro, worksheet
 - Sakai (Forum for “Broader CS Issues”)
 - Create Web page
 - Use jEdit (Text Editor), IDLE
 - Use Python interpreter in interactive mode
 - Write Python programs

**A lot of stuff
but doable!**

Jan 11, 2011

Sprenkle - CSCI111

1

Intro to Labs

- Introduce Student Assistant
 - Amy Clayton '13
- 3 hours to get started on labs
 - Often will need to finish lab after lab period
 - Lab assignments are the majority of your homework
 - Use this lab (P405), preferably, or P413

Jan 11, 2011

Sprenkle - CSCI111

2

Operating Systems

- Manage hardware resources
- Three popular operating system variations:

Mac

PC/Windows

UNIX

- Compare in terms of cost, popularity, available software, security
- Learn Linux (a UNIX variation) in this class
 - “PC” for Windows is a misnomer because all of these OSs are for “personal computers”.
 - Macs are built on UNIX → can use UNIX commands

Jan 11, 2011

Sprenkle - CSCI111

3

P405 Machines

- Run both Linux and Windows
 - Linux natively
 - Windows virtually
 - Takes a while to start up
 - If need to switch, restart. By default → Linux
- Computer should be in Linux
 - If not, tell someone or move to another computer
- P413: Linux-only

Jan 11, 2011

Sprenkle - CSCI111

4

Pause While You Log In

- Follow handout’s instructions
- Open browser
- Navigate to Lab 0, from course’s “Schedule” page
 - We’re starting on the first objective “Learning to Use the Linux Machines” on paper
 - Return to Web page for rest of lab

How different is the User Interface (UI) than Windows or Mac?

Jan 11, 2011

Sprenkle - CSCI111

5

Intro to UNIX Handout

- A lot of words
- Not that difficult
- Introduces terminology and techniques that will be second nature to you in a few weeks
- Periodically check with your neighbor
 - If answers aren’t similar, see TA or me

Jan 11, 2011

Sprenkle - CSCI111

6

Intro to UNIX

- Can execute operations by typing commands in terminals or using GUIs
 - We will use terminals most of the time
 - Today: learn essential UNIX commands
 - Why?
 - Faster to use keyboard than mouse
 - Easier to automate
- File structure
 - Organize our files
 - Hierarchy of *directories* ("folders" in Windows world)

Jan 11, 2011

Sprengle - CSCI111

7

UNIX is a Bad Parent

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

```
sprengle@spartacus Desktop$ mv lab00.ppt.pdf lab00.pdf
sprengle@spartacus Desktop$
```

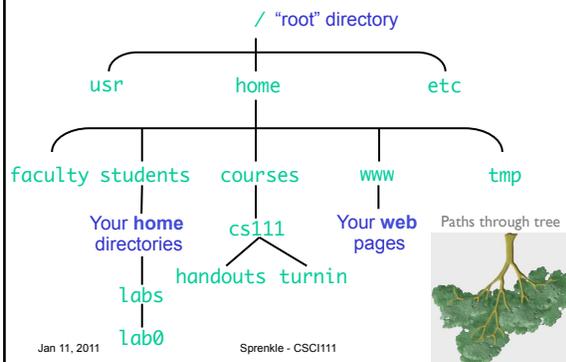
Since you didn't get an error message,
that's correct!

Jan 11, 2011

Sprengle - CSCI111

8

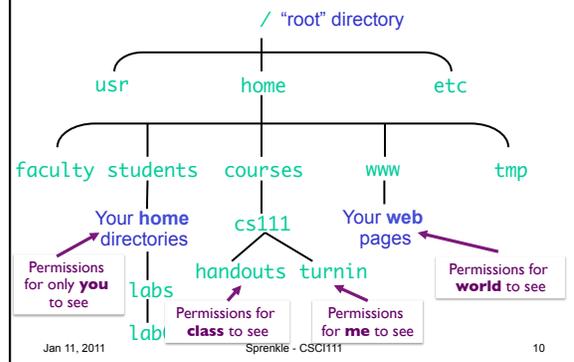
(Partial) Linux File Structure



Jan 11, 2011

Sprengle - CSCI111

(Partial) Linux File Structure



Jan 11, 2011

Sprengle - CSCI111

10

Synchronizing ...

- Everyone started Linux worksheet
- Review:
 - 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.
- Open a new terminal using your shortcut in the top bar

Jan 11, 2011

Sprengle - CSCI111

11

Intro to UNIX: Essential Commands

- Manipulating Files

Command	What it does
ls	list the files, directories in a directory
mkdir	make a directory
cp	copy a file/directory
mv	move a file/directory
rm	remove (delete) a file/directory

- Navigating Directories

pwd	"print working directory"
cd	"change directory"

Jan 11, 2011

Sprengle - CSCI111

12

Python

- A common *interpreted* programming language
 - Runs on many operating systems
- First released by Guido van Rossum in 1991
- Named after *Monty Python's Flying Circus*
- Minimalist syntax, emphasizes *readability*
- Flexible, fast, useful language
- Used by scientists, engineers, systems programmers

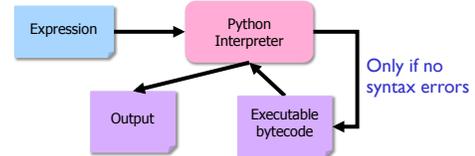
Jan 11, 2011

Sprenkle - CSCI111

19

Python Interpreter

1. Validates Python programming language expression(s)
 - Enforces Python syntax rules
 - Reports syntax errors ← Have a lot of these early on!
2. Executes expression(s)



Jan 11, 2011

Sprenkle - CSCI111

20

Two Modes to Execute Python Code

- **Interactive:** using the interpreter
 - Try out Python expressions
- **Batch:** execute *scripts* (i.e., files containing Python code)
 - What we'll write usually

Jan 11, 2011

Sprenkle - CSCI111

21

Interactive Mode

```
Python 2.4.4 (41, Oct 18 2006; darwin)
[GCC 4.0.1 (Apple Computer, Inc. build 5341)] on darwin
Type "help()", "copyright()", "credits()" or "license()" for more information.
>>> 3
3
>>> 4*5
20
>>> 1-7
-6
>>> word
word
>>> word
NameError: name 'word' is not defined
>>> print 4*5
20
>>> 
```

Jan 11, 2011

Sprenkle - CSCI111

22

Your Turn in Interactive Mode...

- Run the Python interpreter in the terminal
 - `python`
- Enter the following expressions and see what Python displays:
 - `3`
 - `4 * -2`
 - `-1+5`
 - `2 +`
 - `print "Hello!"`
- To quit the interpreter, use Control-D

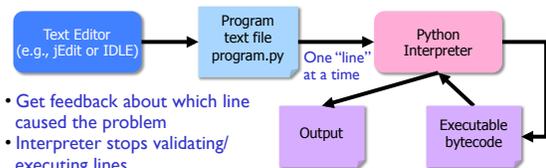
Jan 11, 2011

Sprenkle - CSCI111

23

Batch Mode

1. Programmer types a *program/script* into a *text editor* (jEdit or IDLE).
2. An *interpreter* turns each expression into *bytecode* and then executes each expression



Jan 11, 2011

Sprenkle - CSCI111

24

Example Python Script

```
Text file named: hello.py
# by Sara Sprenkle
# Last modified: 01/11/2011
print "Hello, world!"
```

Print statement

- What does this program do?
 - Validate your guess by executing the program
 - Go into labs/lab0 directory
 - `python hello.py`

Jan 11, 2011

Sprenkle - CSCI111

25

Example Python Script

```
# Program that prints out "Hello, world!"
# by Sara Sprenkle
# Last modified: 01/11/2011
print "Hello, world!"
```

Documentation
-- good style

- Only `Hello, world!` is printed out
- Python ignores everything after the "#"
 - Known as "**comments**" or, collectively, as **documentation**
- Your program should *always* start with a high-level description of what the program does, your name, and the date the program was written

Jan 11, 2011

Sprenkle - CSCI111

26

Review: Executing Python

- Interactive Mode
 - Try out expressions
 - `python`
- Batch Mode
 - Execute Python scripts
 - `python <pythonscript>`

Jan 11, 2011

Sprenkle - CSCI111

27

Practice in Interactive, Batch Modes

- Open the IDLE development environment
 - Command: `idle &`
 - `&` Runs command in "background" so you can continue to use the terminal

Knowing our programming language is named after Monty Python, any ideas about what the development environment is named after?

Jan 11, 2011

Sprenkle - CSCI111

28

Demonstrate Using IDLE

Jan 11, 2011

Sprenkle - CSCI111

29

Honors System: Rules of Thumb

- Discussion of problems/programs - OK
 - Clarification questions
 - Algorithm discussion (on paper, board)
- Debugging help
 - Programmer always "owns" keyboard, mouse
 - Helper can read other's program/debug/help, up to 5 minutes
 - Ask TA or me or email me for problems that require more time

Jan 11, 2011

Sprenkle - CSCI111

30

Lab 0 Checklist

- Linux Worksheet
- Sakai access
- Web Page
- Python practice, programs
- Print lab
- Pledge ***all*** work you submit