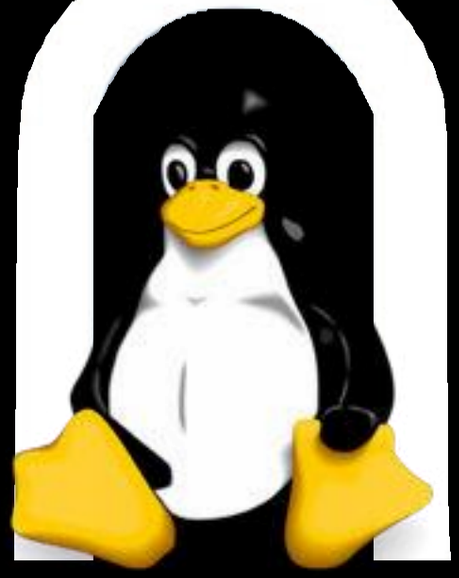
Introduction To

UNIX



Hui Jiang
jianghui@umich.edu

Slides credit: Barry Grant (bjgrant@ucsd.edu)

Introduction to Biocomputing

Monday	**Introduction to UNIX**
Tuesday	**Introduction to Programming with Python**
Wednesday	**Data Analysis and Graphics with R**
Thursday	**Version Control & Cluster Computing**
Friday	**Group Projects**

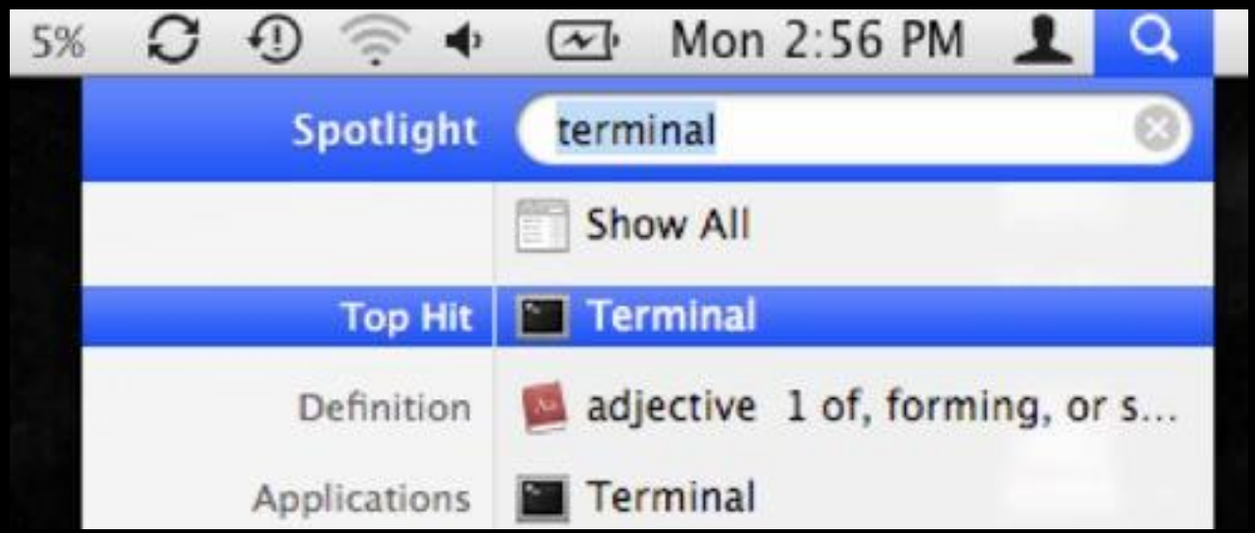
Todays Menu

	Time	Topics
I	9:00-10:15 AM	**Setup and Motivation**
	10:15-10:30 AM	Coffee Break
II	10:30-12:00 AM	**Beginning Unix**
III	12:00-1:00 PM	Lunch
	1:00-2:15 PM	**Working with Unix**
	2:15-2:30 PM	Coffee Break
IV	2:30-4:00 PM	**How to Get Working**

Lets get started…

Do it Yourself!

Mac
Terminal



PC
MobaXterm

Setup Checklist

Questionnaire

- **Mac**: Terminal *or* **PC**: MobIXterm
- **Mac**: Git install *or* **PC**: MobIXterm git & CygUtils plugins
- Python Anaconda install
- R and RStudio install
- Cluster access form submitted and Duo mobile app obtained
- Example data downloaded: **http://tinyurl.com/day1-unix**

```
# In your terminal type
> which git
> git --version
```

Motivation

Why do we use Unix?

Modularity	Core programs are modular and work well with others
Programmability	Best software development environment
Infrastructure	Access to existing tools and cutting-edge methods
Reliability	Unparalleled uptime and stability
Unix Philosophy	Encourages open standards

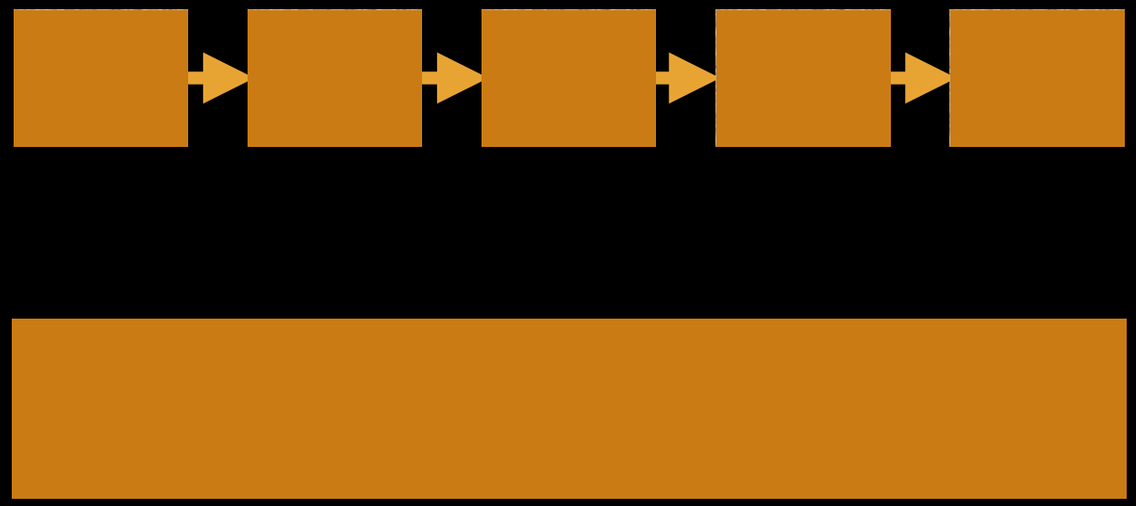
Modularity	Core programs are modular and work well with others
Programmability	Best software development environment
Infrastructure	Access to existing tools and cutting-edge methods
Reliability	Unparalleled uptime and stability
Unix Philosophy	Encourages open standards

Modularity

The Unix shell was designed to allow users to easily build complex workflows by interfacing smaller **modular programs** together.

An alternative approach is to write a **single complex program** that takes raw data as input, and after hours of data processing, outputs publication figures and a final table of results.

Which would you prefer and why?



Modular

vs

Custom

Advantages/Disadvantages

The ‘monster approach’ is customized to a particular project but results in massive, fragile and difficult to modify (therefore inflexible, untransferable, and error prone) code.

With **modular workflows**, it’s easier to:

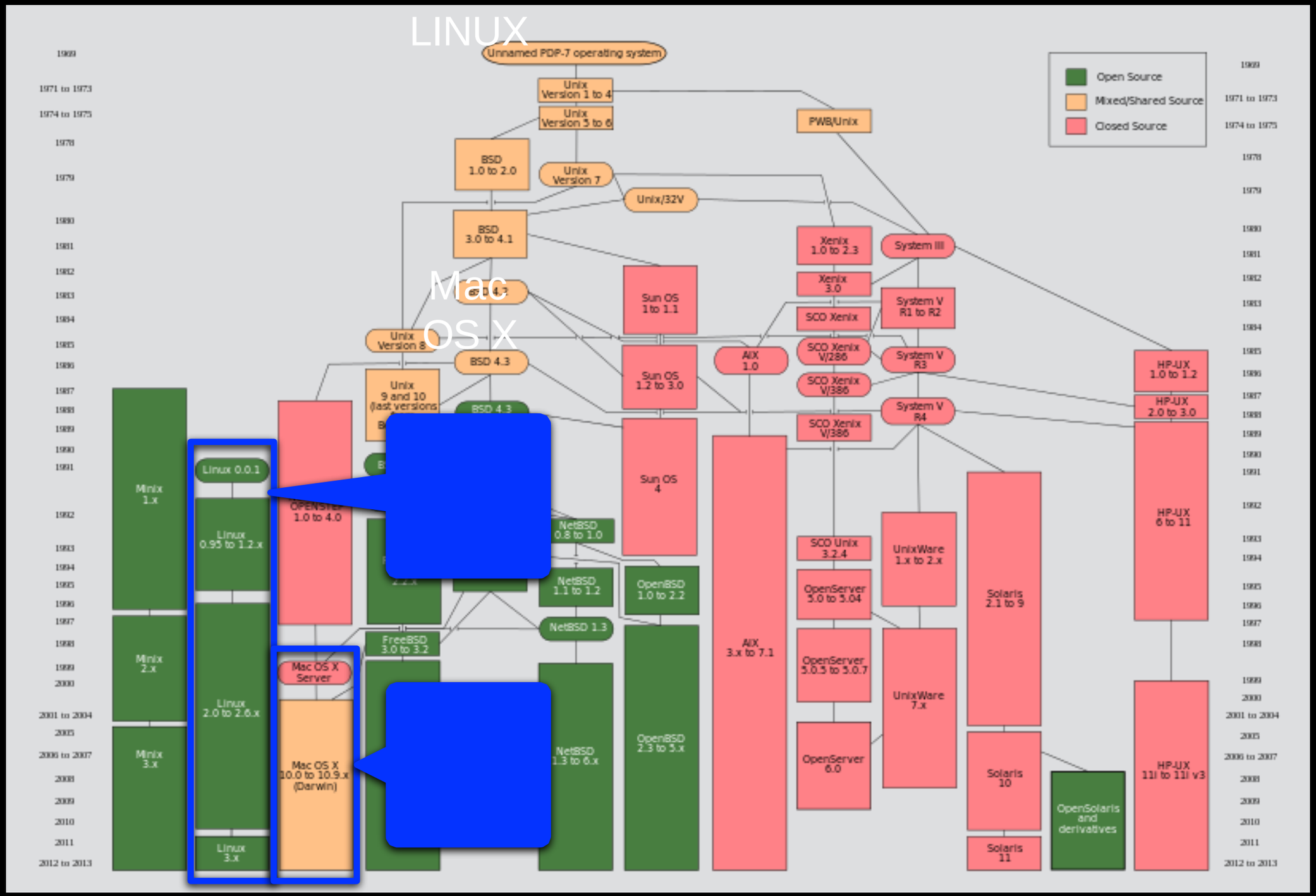
- Spot errors and figure out where they’re occurring by inspecting intermediate results.
- Experiment with alternative methods by swapping out components.
- Tackle novel problems by remixing existing modular tools.

Unix ‘Philosophy’

“Write programs that do one thingand do it well. Write programs to work together and that encourage open standards. Write programs to handle text streams, because that is a universal interface.”

— Doug McIlory

Unix family tree [1969-2010]



Source: https://commons.wikimedia.org/wiki/File:Unix_history-simple.svg

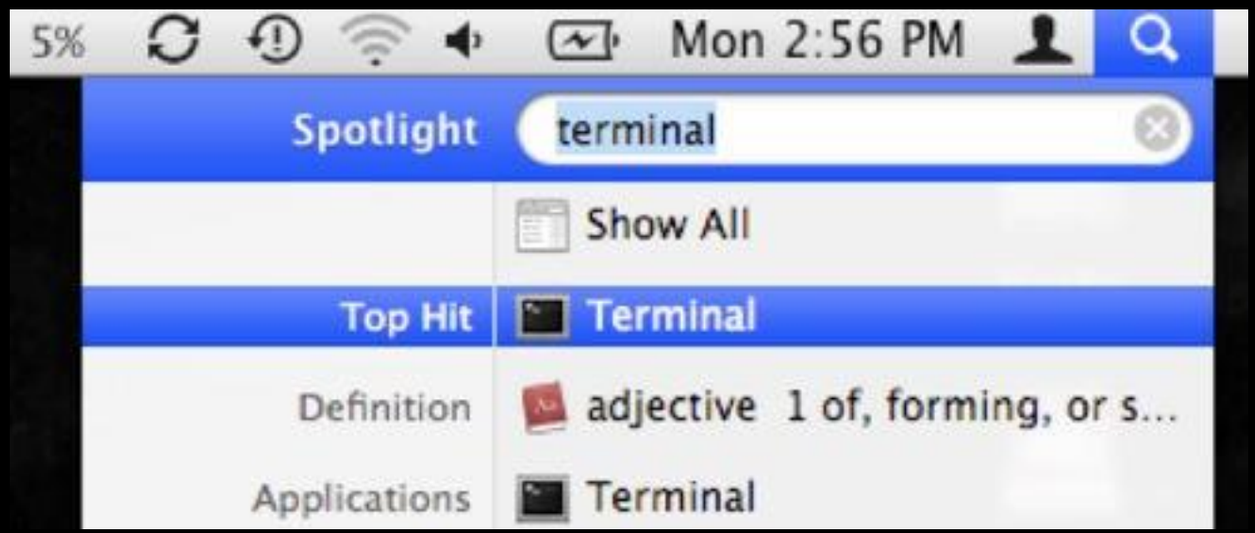
Basics	File Control	Viewing & Editing Files	Misc. useful	Power commands	Process related
ls	mv	less	chmod	grep	top
cd	cp	head	echo	find	ps
pwd	mkdir	tail	wc	sed	kill
man	rm	nano	curl	uniq	Crl-c
ssh	| (pipe)	touch	source	git	Crl-z
	> (write to file)		cat	R	bg
	< (read from file)		tmux	python	fg

Basics	File Control	Viewing & Editing Files	Misc. useful	Power commands	Process related
ls	mv	less	chmod	grep	top
cd	cp	head	echo	find	ps
pwd	mkdir	tail	wc	sed	kill
man	rm	nano	curl	uniq	Crl-c
ssh	| (pipe)	touch	source	git	Crl-z
	> (write to file)		cat	R	bg
	< (read from file)		tmux	python	fg

Lets get started…

Do it Yourself!

Mac
Terminal



PC
MobaXterm

Test: Connecting to remote machines (with **ssh**)

- Most high-performance computing (HPC) resources can only be accessed by **ssh** (Secure SHell)

```
> ssh [user@host.address]
> ssh jianghui@scs.dsc.umich.edu
```

Test: Your software versions

- We will use the **which** command to locate your versions of the major software we will be using this week.
  - > which **R**
  - > **R** --version

  Now do the same for **python** and **git** , *i.e.*
  - > which **git**
  - > **git** --version
- If you get an ‘error’ or ‘not found’ msg let us know!