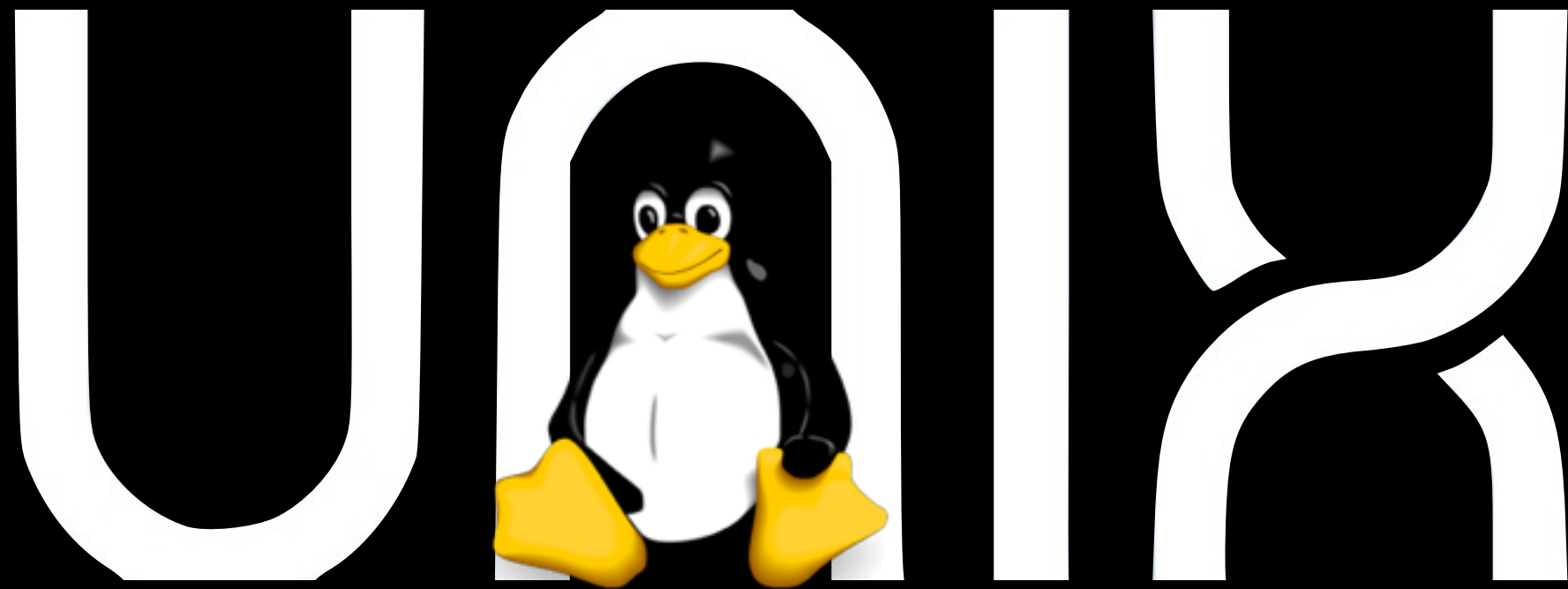


Introduction To



Barry Grant

bjgrant@umich.edu

<http://thegrantlab.org>

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

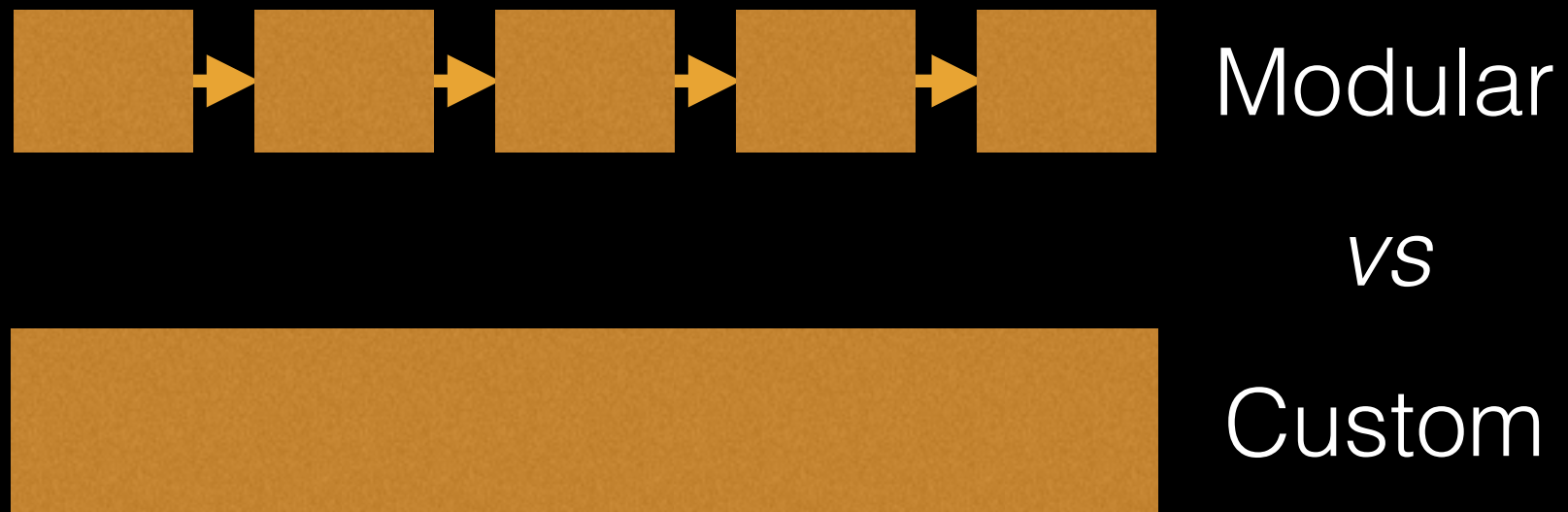
The Unix shell was designed to allow users to easily build complex workflows by interfacing smaller **modular programs** together.



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?



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.

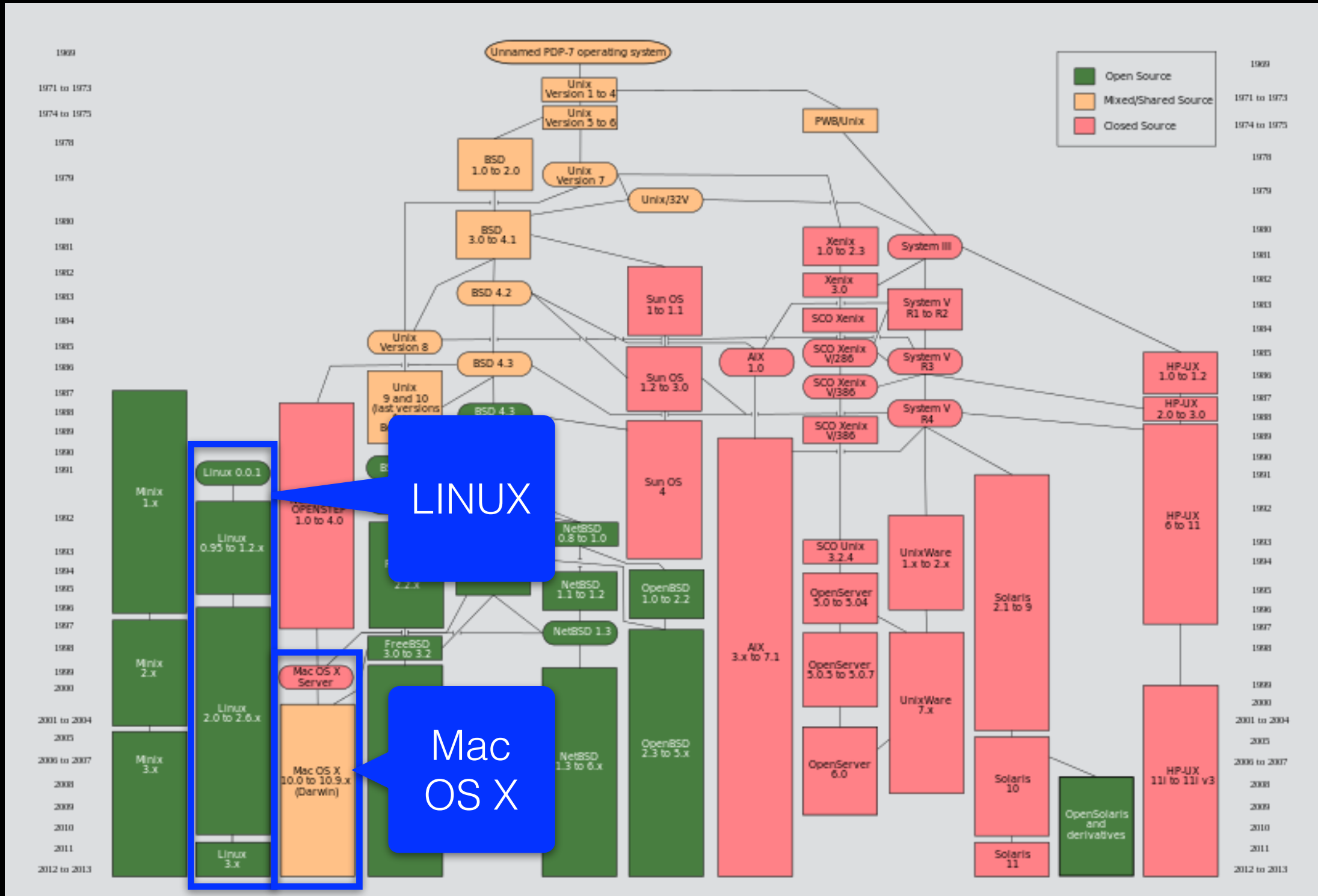
Unix ‘Philosophy’

“Write programs that do one thing and 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)			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	Ctrl-c
ssh	 (pipe)	touch	source	git	Ctrl-z
	> (write to file)		cat	R	bg
	< (read from file)			python	fg

Lets get started....

Do it Yourself!

