

L03 - MONDAY, AUGUST 28  
CPSC 1070 | FALL 2023

# INTRO TO UNIX CONTINUED + INTERPRETING READINESS SCORE



*School of*

**COMPUTING**

*Clemson University*

# ANNOUNCEMENTS

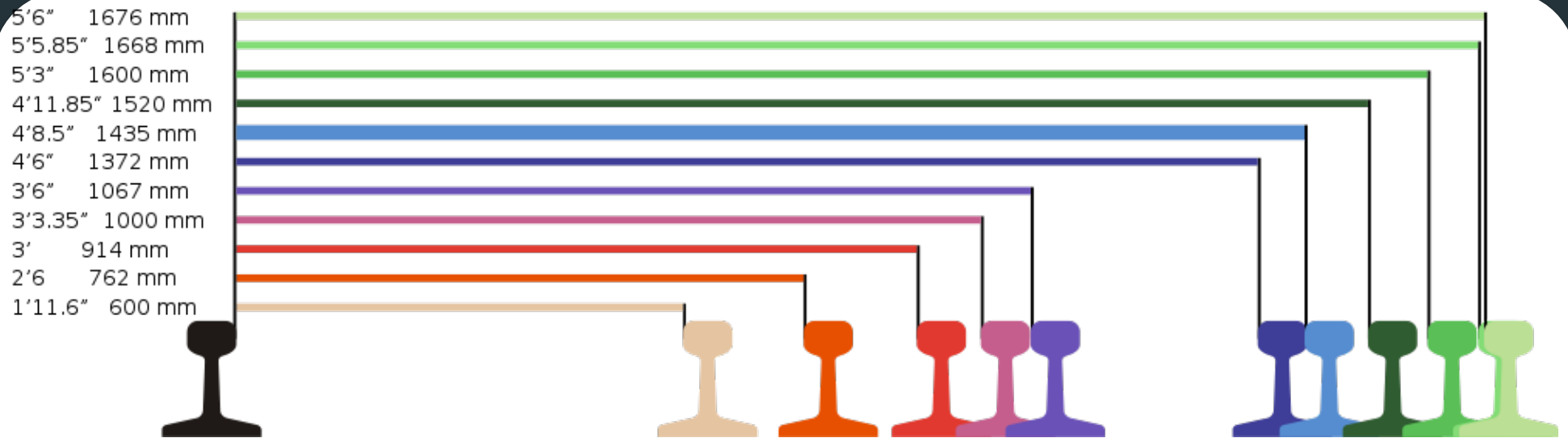
- HW-0 is assigned on Canvas and due on Friday
- Labs continue today
  - Note: You will be assigned to a random cohort within lab on Wednesday
- Reminder: Teams has the following:
  - A "CPSC 1070 Team" with multiple channels available to ask questions and is monitored by the instructional staff
  - Ability to direct message teaching staff during office hours
  - Don't send email when you can use Teams!

# LEARNING OUTCOMES

BY THE END OF THIS CLASS, YOU SHOULD BE ABLE TO:

- List benefits of a UNIX operating system
- Describe what a UNIX shell is
- Use a UNIX terminal to connect to the School of Computing
- Use basic UNIX navigations within a terminal
- Compare and contrast direct manipulation versus a command line interface

# STANDARDIZATION



# UNIX STANDARDIZATION

- 1989: **ANSI** Standard X3.159-1989 for the C programming language
  - defines syntax and semantics
  - **standard library** <- very important
- 1999: C-standard was updated and approved
- (ISO/IEC 9899:1999)
  - not many changes to the **POSIX standard** (restrict keyword)

# UNIX VARIANTS

- Is Mac OS X a UNIX system?

Yes <http://arstechnica.com/apple/news/2007/08/mac-os-x-leopard-receives-unix-03-certification.ars>

- Is FreeBSD a UNIX system?

Not Yet

- Is Linux a UNIX system?

Not Yet

- Is Solaris a UNIX system?

Yup

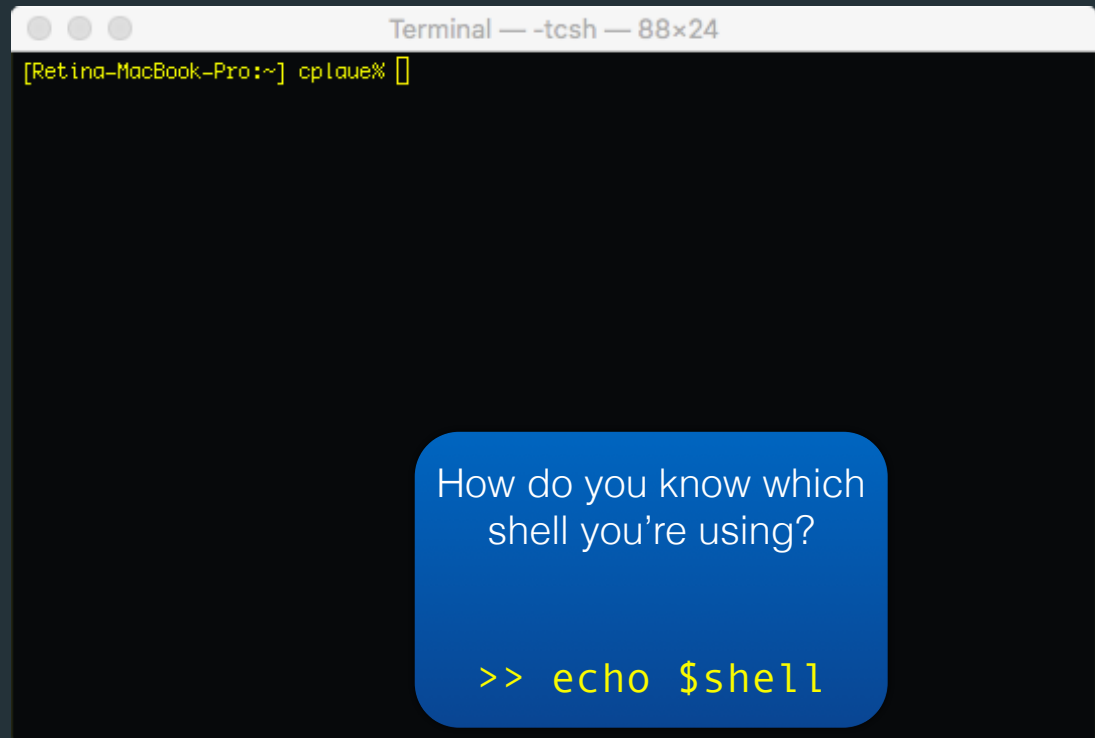


# UNIX TODAY

- Unix supports many users running many programs at the same time, all sharing the same computer system
- Unix supports information sharing
- Geared towards facilitating the job of creating new programs
- Examples: SunOS and Solaris, GNU: Linux, Apple: OS X (Darwin), IRIX: FreeBSD
- Android???

# SHELLS: THE TRADITIONAL UNIX INTERFACE

- A **shell** is a program that starts automatically when you login and provides a **command line interface** between the human and computer
- Uses a command language
- Multiple versions of shells:
  - sh** Bourne shell
  - ksh** Korn shell
  - cs** C Shell
  - tcsh**
  - bash**



A terminal window titled "Terminal — -tcsh — 88x24" is shown. The prompt is "[Retina-MacBook-Pro:~] cplaue%". A blue callout box contains the text "How do you know which shell you're using?" and the command ">> echo \$shell".

```
Terminal — -tcsh — 88x24
[Retina-MacBook-Pro:~] cplaue%

How do you know which
shell you're using?

>> echo $shell
```



## Mac terminal

```
Terminal — tcsh — 80x24

[MacBook-Pro:~] cplaue% ssh plaue@odin.cs.uga.edu
plaue@odin.cs.uga.edu's password:
Last login: Thu Oct 22 15:20:26 2009 from h70-33-70-123.paws.uga.edu
-bash-3.2$ which java
/usr/bin/java
-bash-3.2$ java -version
java version "1.6.0_17"
Java(TM) SE Runtime Environment (build 1.6.0_17-b04)
Java HotSpot(TM) 64-Bit Server VM (build 14.3-b01, mixed mode)
-bash-3.2$ ls -l
total 32
drwxr-xr-x 4 plaue users 4096 Aug 20 12:01 class_demo
drwxr-xr-x 2 plaue users 4096 Nov  9 17:16 Desktop
drwxr-xr-x 3 plaue users 4096 Aug 20 10:51 temp
drwxr-xr-x 2 plaue users 4096 Sep 30 14:05 temp2
-bash-3.2$ exit
logout
Connection to odin.cs.uga.edu closed.
[MacBook-Pro:~] cplaue% 
```

# BASIC UNIX COMMANDS

- We'll walk through some basic UNIX commands in class today
- You'll get experience using basic UNIX commands during CoLab 01
- Additional resources  
<http://www.ee.surrey.ac.uk/Teaching/Unix/>
- Before we begin.....

If you've never used a command-line interface before, you might find learning one a bit painful....



# USER INTERFACE HISTORY: DIRECT MANIPULATION

- Many of the interfaces you use are examples of **direct manipulation**
- In 1982, Shneiderman coined this term describing the appeal of a rapidly-developing graphically-based interaction including:
  - object visibility
  - incremental actions and rapid feedback
  - reversibility encourages exploration
  - replaces language with actions
  - syntactic correctness of all actions
- **WYSIWYG** & Apple Macintosh



# USER INTERFACE SCIENCE: LEARNING & METAPHORS

- All computer use is problem-solving or learning to some extent
- Relating computing to real-world activity is effective learning mechanism
- Examples
  - File management on office desktop
  - Financial analysis as spreadsheet
- Tension between literalism and magic

# USER INTERFACE SCIENCE: RECALL VS RECOGNITION

- The **WIMP interface** is using recognition by providing visible actions that a user can perform
  - You can click icons, use menus, etc to perform commands
- In a **command-line interface**, you type out commands to the computer at a **prompt**

# GENERAL UNIX COMMAND FORMAT

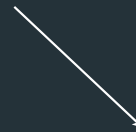
\$ command -options targets<return>



Verb

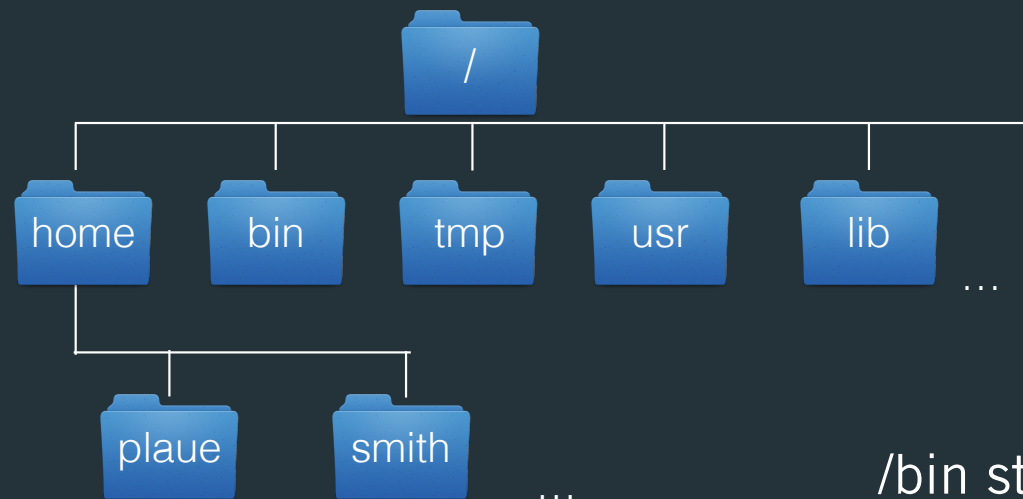


Adverb



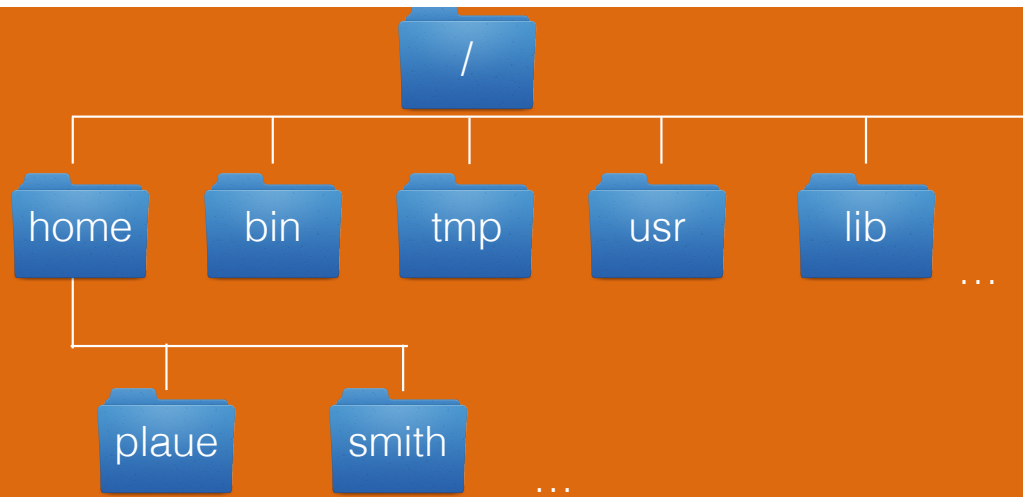
Target

# FILESYSTEM: HOW FILES ARE STORED



/bin stands for binaries  
/lib stands for c libraries  
/dev stands for devices  
/tmp stands for temp files





What are the basic types of actions you'd want to take in a filesystem?

# ACTIVITY

# ESSENTIAL UNIX COMMANDS

- `ls` list contents of a directory
- `cd` change directory
- `pwd` show the current working directory
- `man` show help "manual" page
- `passwd` changes your password
- `echo` "echo" a string to standard output

# ESSENTIAL UNIX COMMANDS

- `mkdir`            create a directory
- `rmdir`            removes a directory
- `touch`            create a file
- `rm`                remove/delete a file
- `cp`                copies a file
- `mv`                moves or renames a file

# ESSENTIAL UNIX COMMANDS

- `cat` lists a file's contents
- `which` lists the complete path of a command
- `less` scroll through a file
- `quota -v` see how much space you have available
- `exit` logs you off
- `ps` shows the current `processes` running
- `kill` kills a process  
look up a process's id with `ps` first

# CAUTION ABOUT KILL

- You must issue kill commands to stop processes (think programs) that are taking up a lot of resources (or you will tick off system support)

- Example:

```
kill 12778
```

```
kill -9 12778
```



# HELPFUL UNIX KEYSTROKES

- control + z      stop/suspend the current command
  - control + c      kills some commands or processes
  - tab              autocompletes commands file names
  - up/down keys    scroll through command history
- } Try these before k i l l

SO, HOW DO YOU FEEL?



You probably feel a bit overloaded.

That's because you haven't mastered the learning that enables recall!

You're probably wondering why anyone would want to use a command-line interface?

Millions of people do every day!

Why?



# PRACTICE BRINGS COMFORT

- You can easily connect to one of the School of Computing's Linux computers via your own personal laptop or desktop computer!
- Save this bookmark as a helpful resource:  
<https://www.computing.clemson.edu/help/unixaccount.html>)

# LINUX AT CLEMSON'S SCHOOL OF COMPUTING

\*\*\*\*\*

Welcome to the  
CLEMSON UNIVERSITY SCHOOL OF COMPUTING  
Unauthorized use is prohibited!

## PARTIAL LIST OF PUBLIC CLIENT MACHINES

Hostnames	Location	OS	Architecture	Mem
joey1..18	n/a	Ubuntu 20.04	Core(TM) i7-4790	32GB
cerf1..30	McAdams 110D	Ubuntu 20.04	Core(TM) i5-7500	16GB
ada1..15	McAdams 110B	Ubuntu 20.04	Core(TM) i5-7500	16GB
cirrus1..9	n/a	Ubuntu 20.04	Xeon(R) W-2255	64GB
newton	n/a	Ubuntu 20.04	Xeon(R) E5645	96GB

\*\*\*\*\*

- \* Off-campus SSH access: [access.computing.clemson.edu](https://access.computing.clemson.edu)
- \* General information can be found at [computing.clemson.edu/help](https://computing.clemson.edu/help)
- \* Virtual Linux Desktop: [virtual.computing.clemson.edu](https://virtual.computing.clemson.edu)
- \* Questions or problems with Linux lab systems? Email [ithelp@clemson.edu](mailto:ithelp@clemson.edu)

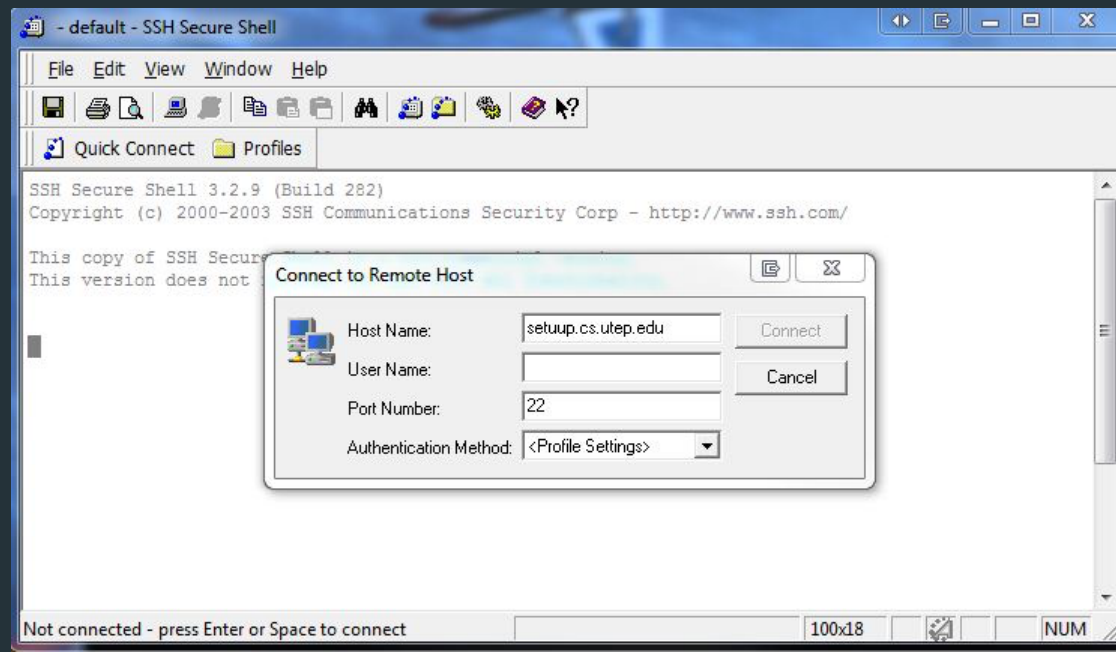
\*\*\*\*\*

# GAINING ACCESS TO LINUX AT S.O.C.

- Go to a School of Computing Lab (e.g. McAdams 110A, 110d, or Barre B108)
  - log in using your Clemson user ID and Clemson password
- Using your personal computer, use a [Secure Shell \(SSH\) program](#) to connect remotely
- On campus:
  - babbage1.[computing.clemson.edu](#) thru babbage34.computing.clemson.edu
- Off campus:
  - access.[computing.clemson.edu](#) (or access 1 or access2) to initially connect to campus, then make a ssh connection to a Babbage computer as listed above

# SSH CLIENT

- A software program that allows you to securely create a shell window to a remote computer/server
- SSH for Windows CCIT Link: [http://www.clemson.edu/ccit/software\\_applications/software/web\\_downloads.html](http://www.clemson.edu/ccit/software_applications/software/web_downloads.html)



# SSH ON A MAC OS X COMPUTER

- Finder > Applications > Utilities > Terminal  
*(you can drag Terminal to the Dock for quick access)*
- Choose the right on/off campus connection and run the following command:
- Where `ada1` is your choice of one of the SoC computers:
  - On campus: `username@ada1.computing.clemson.edu`
  - Off campus:
    - `username@access1.computing.clemson.edu` (Step 1)
    - `username@ada1.computing.clemson.edu` (Step 2)

# PROGRAMMING LANGUAGES



- **High-level languages:** write expressions that look sort-of like English sentences
- **Assembly languages:** English-like abbreviations to represent elementary operations
- **Machine language:** generally consists of numbers which are ultimately reduced to 1s and 0s

# READINESS QUIZ

0



20-30	You are likely in good shape for CPSC 1070
15-19	You will need to brush up and come up with a regular plan to practice concepts talked about in class.
0-14	Concerns that you will struggle with the pace of this course and/or have missing foundational knowledge. Seriously consider taking CPSC 1010.

If you'd like to drop-back to CPSC 1010 (CS-1 for those with some programming), there will be a link posted on Canvas

### Considerations and Caveats

- What grade did you earn in 1060 or AP-CS-A? How long ago was that?
- Is this your major and do you want a solid foundation for future courses?
- Dropping back now doesn't put you behind in classes
- Spring 2022
  - 43% in the red group ended up with a grade of DWF
  - 64% in the green group earned an A in the class

