#1.1: Command Line and Navigating Files

Slides with “ARQ” are copied or adapted from Aaron R. Quinlan

Access the slides and files here:

https://github.com/j-berg/bioinformatics_bootcamp

Course Overview and Goals

- Week 1 & 2:
  - Using the command line for bioinformatics
- Week 3 & 4:
  - Basics of programming and writing simple tools in Python
- Week 5 & 6
  - Statistics and visualization in R

Course Overview and Goals

- Purpose of the course is to introduce you to the language and power of programming
- Help you start thinking about how you can use programming and bioinformatics in your research
- Determine which language is most applicable to you to start with

Class structure

- Mondays/Wednesdays
  - Conceptual lecture with demonstrations

- Fridays
  - Review session for weeks sessions
  - Troubleshooting homework problems together

Some words of advice

- Just like learning any new language, programming takes time and can be difficult
- With more practice, fluency will improve
- Don’t be discouraged if you have to Google everything (I still do this daily, even with tasks I’ve done time after time)
- Ask your classmates for help on Slack – troubleshoot together
- Best way to learn is to break the code, troubleshoot it, find out why it didn’t work, and find out why the solution actually works

Pseudo-code

- Using simplified language to plan your program
- Helps keep your programming organized
- Like outlining an essay before writing to keep it concise, focused, and flowing

Pseudo-code

```
"""
Purpose:
- take user input of a number
- check if it is even or odd
- tell user what it is

-----------------------------

Take user input

Check input is a number (and is integer)

If integer, check type

  If odd, print out to user
  If even, print out to user

If not integer, tell user
"""
```

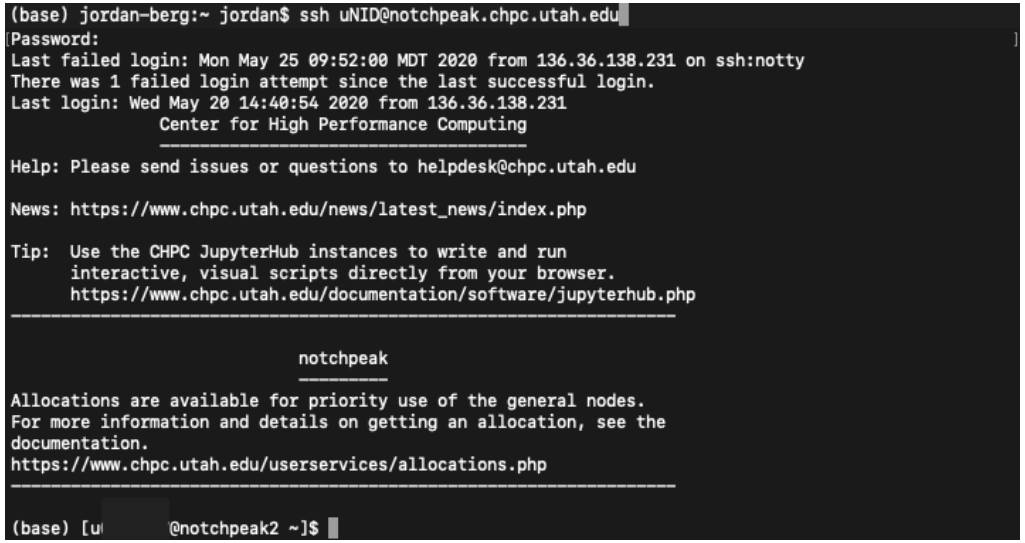
```python
# Take user input
def check_number(n):

  # Check input is a number (and is integer)
  if type(n) == "int":

    # If even, print out to user
    if n % 2 == 0:
      print("Number is even")

    # If odd, print out to user
    else:
      print("Number is odd")

  # If not integer, tell user
  else:
    print("User input is not an integer")
```

Logging onto the supercomputer

- Computational biology tasks are resource intensive
- Super-computing resources commonly need to be used
- Lets you step away while the computer does the hard work
  - Or else you won’t be able to use your computer for a couple of days

macOS

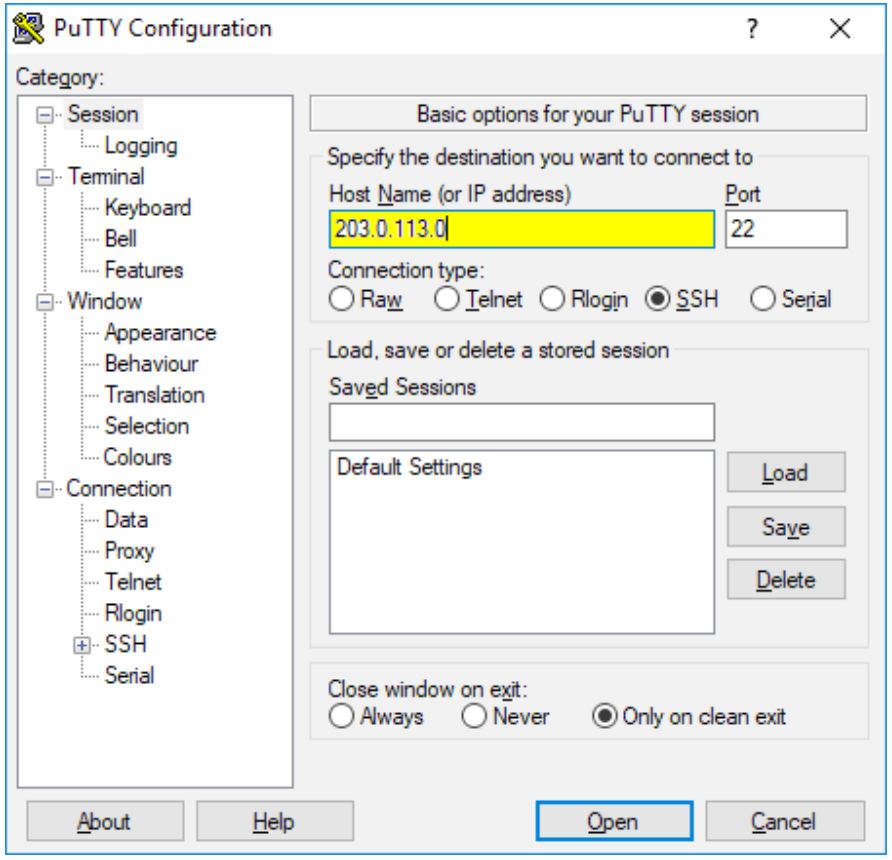
- Open finder
- Type “Terminal” and press enter
- Type the following:



- Verify host key
- Password will not show characters while typing
- To exit, type “logout”

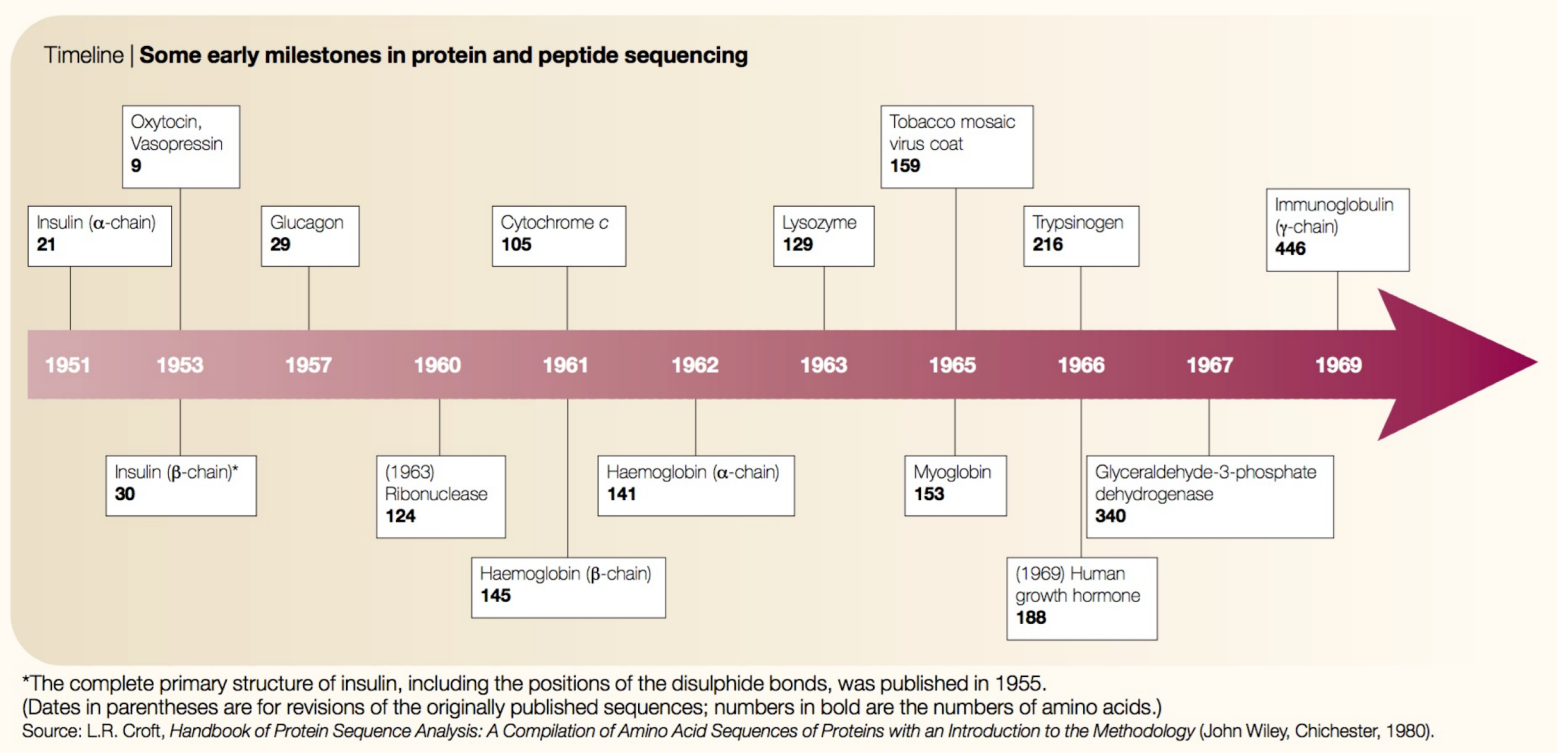
Windows

- Download PuTTY
- “Host Name”: notchpeak.chpc.utah.edu
- “Connection type”: SSH
- Press the ‘Open’ button at the bottom of the dialog box, and PuTTY will begin trying to connect you to the server.
- Verify host key
- Enter the username and the password, and the server should grant you access and begin your session. If you have mistyped your password, most servers will give you several chances to get it right.
- To exit, type “logout”



Comp. biology was born in the 1960s

- Began with amino acid sequencing in 1960's
- Need for computational power to answer questions and study protein biology
- Scarcity of academic computers was no longer a major problem
- Allows analyses that would take several lifetimes by hand



Joel Hagen, "The origins of bioinformatics", NRG, Dec. 2000

What is Unix?

Definition 1: Your interface with the underbelly of the computer

Definition 2: Where computational genomics is done.

Definition 3: Your dear friend.

Recommended reading: “The Evolution of the Unix Time-sharing system”, Dennis M. Ritchie
https://pdfs.semanticscholar.org/f64f/6e66da16e93ebf4221fc8915b2420fd56b66.pdf

Unix history

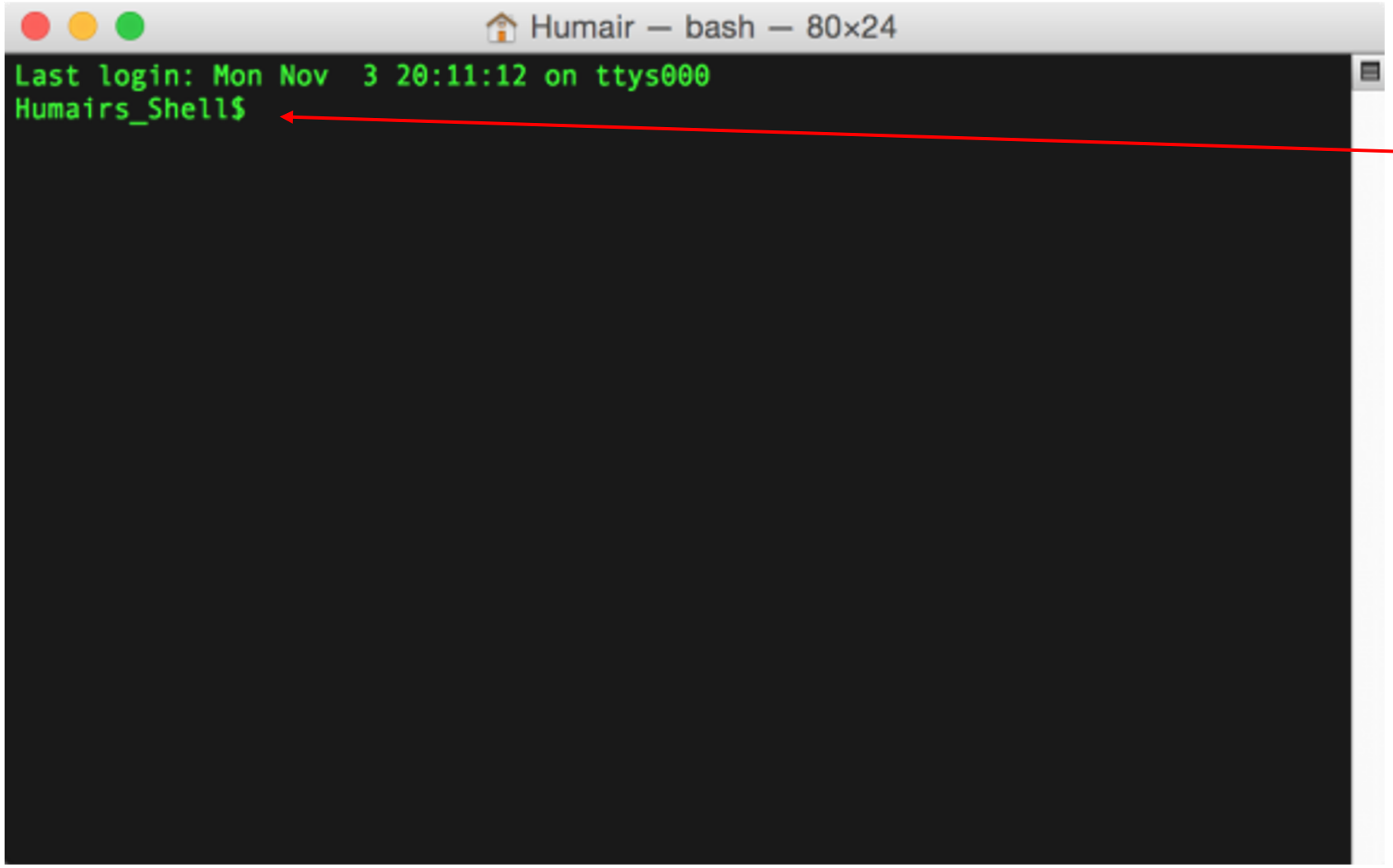
- Initial file system, command interpreter (shell), and process management started by Ken Thompson
- Vast array of simple, dependable tools that each do one simple task.
- By combining these tools, one can conduct rather sophisticated analyses
- Wildly popular platform for high performance computing. Supports parallelism.
- Most computational biology tools come with a command line interface for using the tool.

Ken Thompson (sitting) and Dennis Ritchie working together at a PDP-11

Credit:https://en.wikipedia.org/wiki/History_of_Unix

Unix basics

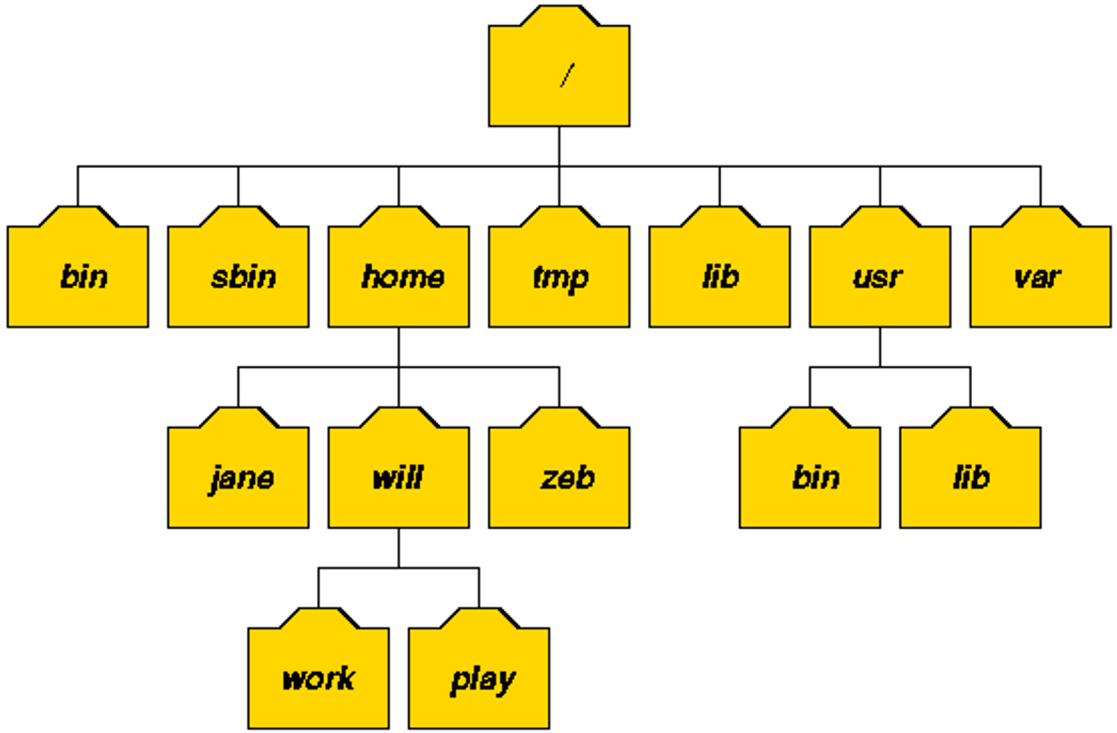
The “prompt”

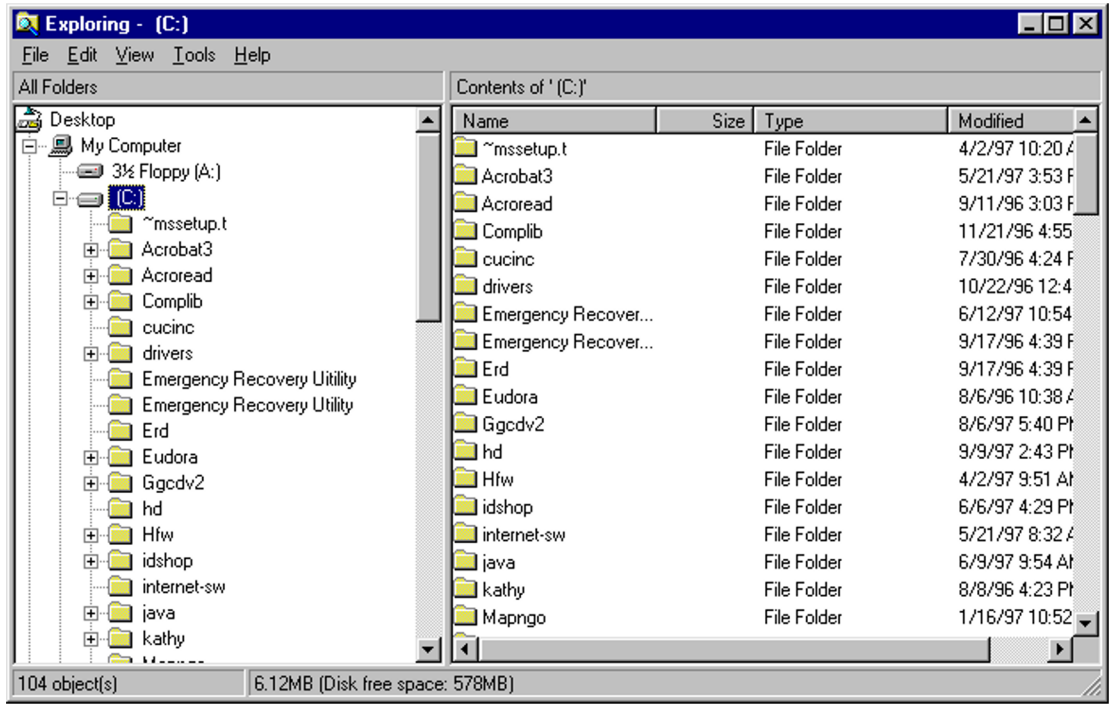


The prompt is just a patient little thing that waits around for you to tell it what to do via “commands”.

Command syntax must be exact. In this way, Unix is dumb. It cannot infer what you meant if you misspell, provide the wrong syntax, etc.

The Unix file system. A tree just like OSX and Windows



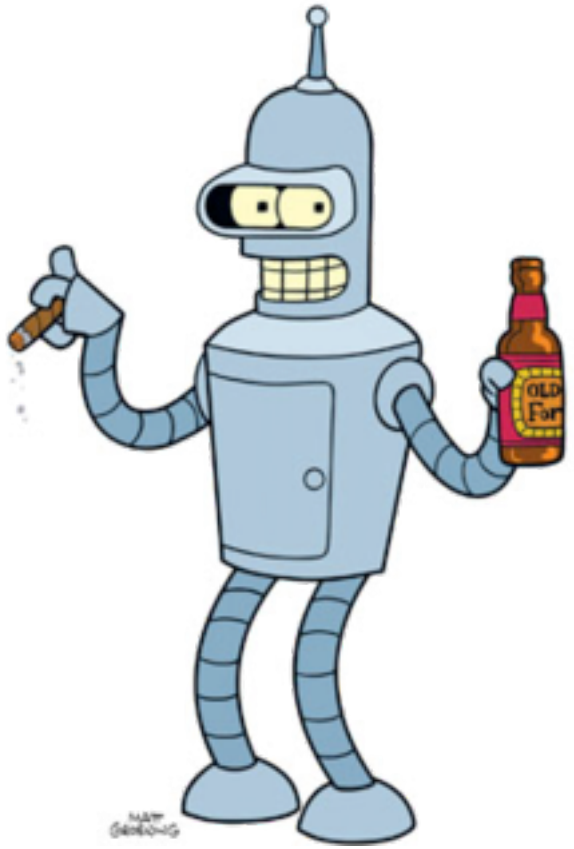


I am lazy. You should be too. Shortcuts!!!

1. Go back to past commands with the arrow keys.
2. The `history` command will report the commands you have used in the past.
3. Type the "Ctrl" "r" keys at the same time to bring up a search for commands that contain search terms. Use the arrow keys to cycle through all commands that match the search term.
4. Use the "Tab" key for autocomplete - just like your smartphone!
5. Type the "Ctrl" "c" keys at the same time to kill a command.

Unix is a stupid robot

- Spaces: Spaces in names of files could be interpreted as a new command, use underscores instead!
- Case sensitivity: If a command of file name is capitalized, it needs to be typed that way!
- Spelling: There is no spell check. If you mis-type a command or file name, it will not find it!



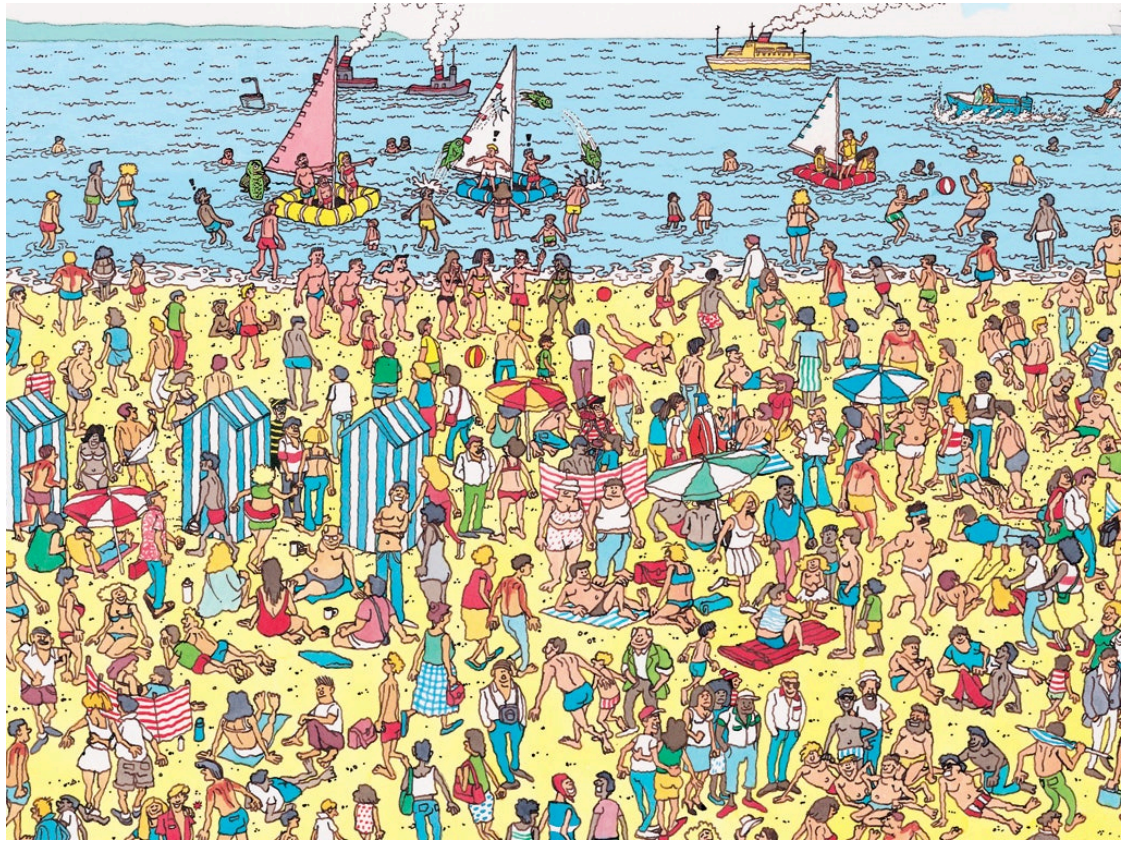
https://en.wikipedia.org/wiki/Bender_(Futurama)#/media/File:Bender_Rodriguez.png

Where am I?

“Print Working Directory”
$ pwd

```
(base) [u       @notchpeak2 test]$ pwd
/uufs/chpc.utah.edu/common/home/u[redacted]/test
```

Will output the absolute pathway to your current working directory



https://www.independent.co.uk/arts-entertainment/books/news/wheres-the-brains-behind-wally-6261459.html

How do I move?

- “Change directory”

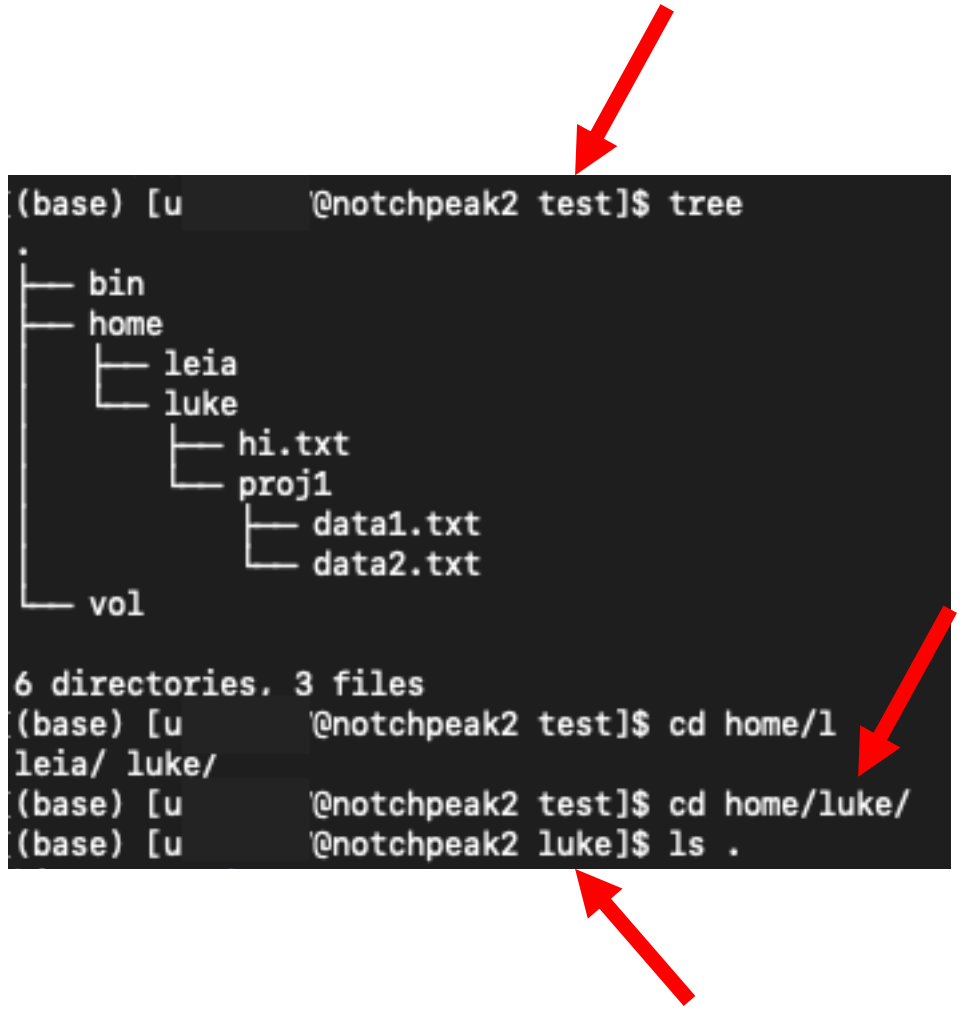
$ cd

Move your current working directory

We are in the “test” directory

We want to move to the “luke” directory

Only the text after the “$” is the command

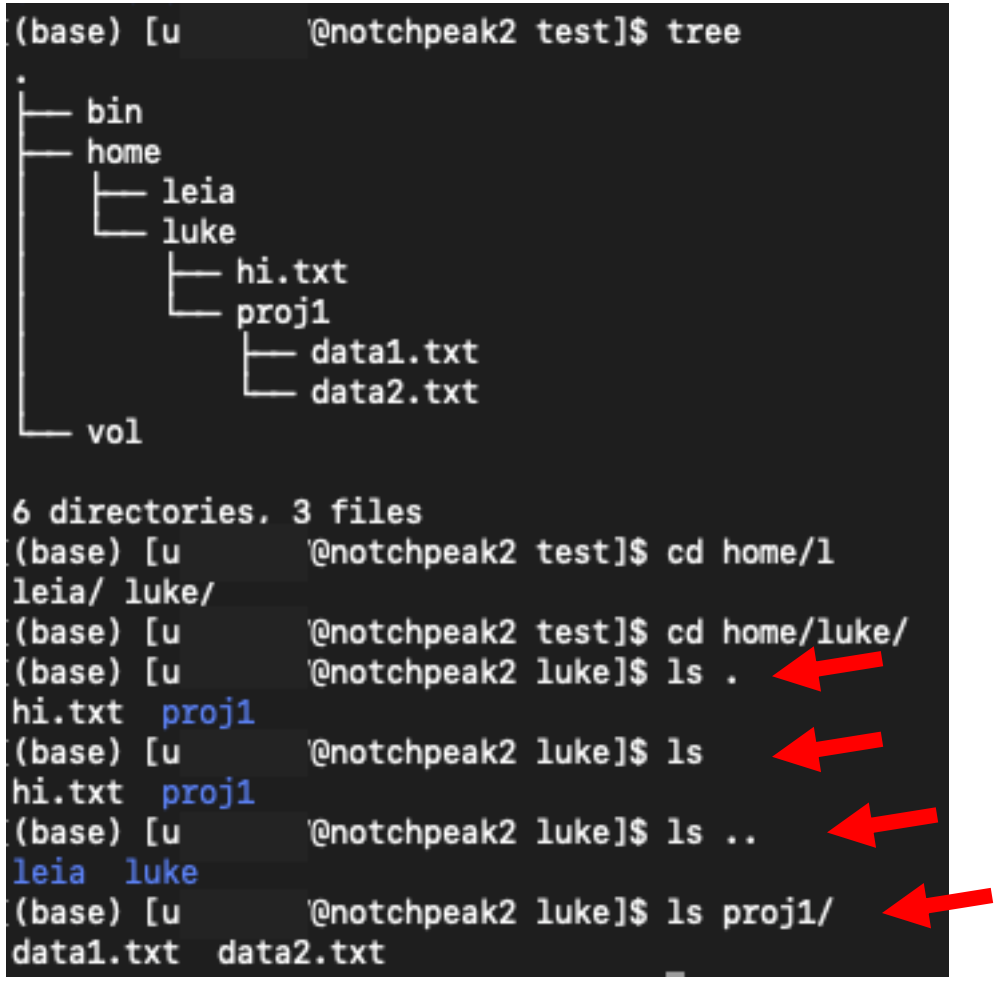


What else is around me?

- List

$ ls

- Lists the files and folders in the specified directory
- . => current directory
- .. => back one directory
- / => separate folders in a folder pathway

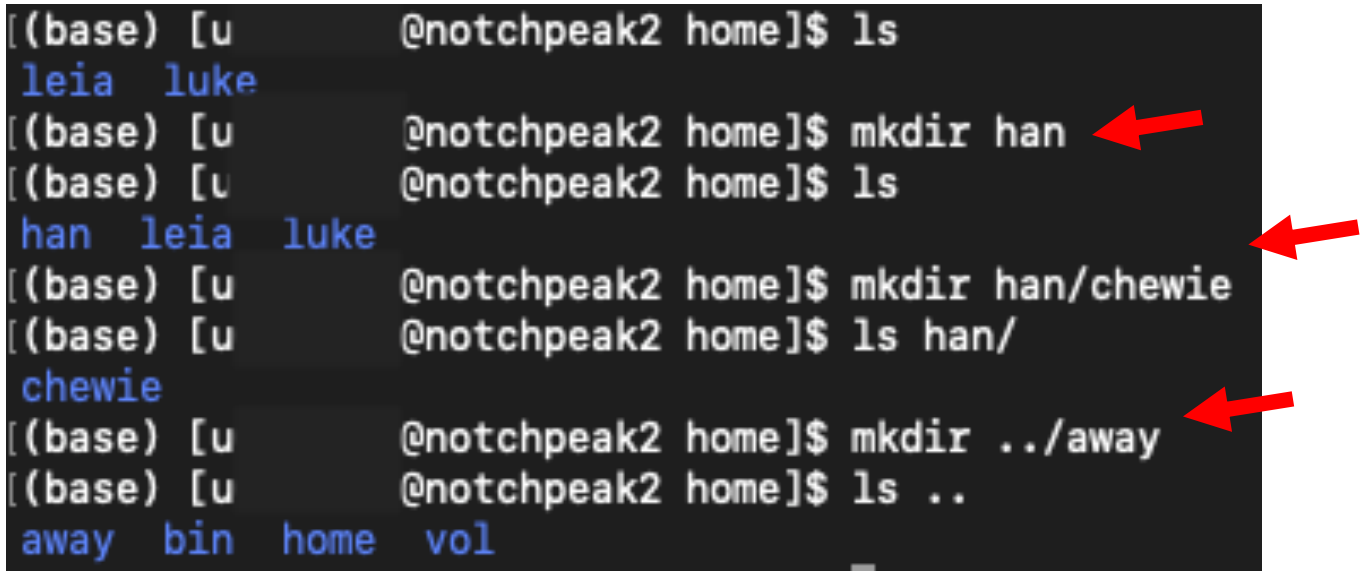


How to I make a new folder?

- “Make directory”

$ mkdir

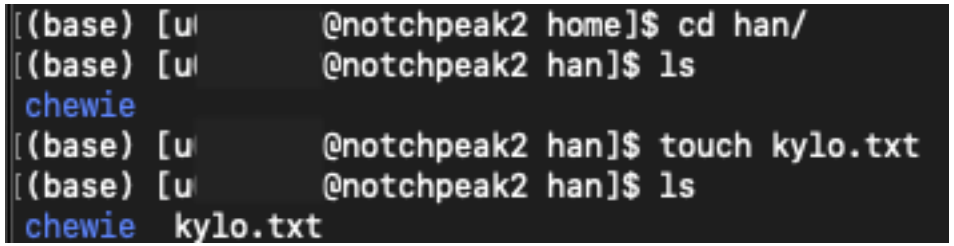
Flexible directioning – can make one in your current directory, in a directory in another folder, or one directory backwards



How do I make a new file?

$ touch

Create a new file, plain and simple



How do I add text to a file?

- > takes the standard output of the command on the left and redirects it to the file on the right.

- >> takes the standard output of the command on the left and ***appends*** it to the file on the right.

How do I read a file?

- “Concatenate”

$ cat

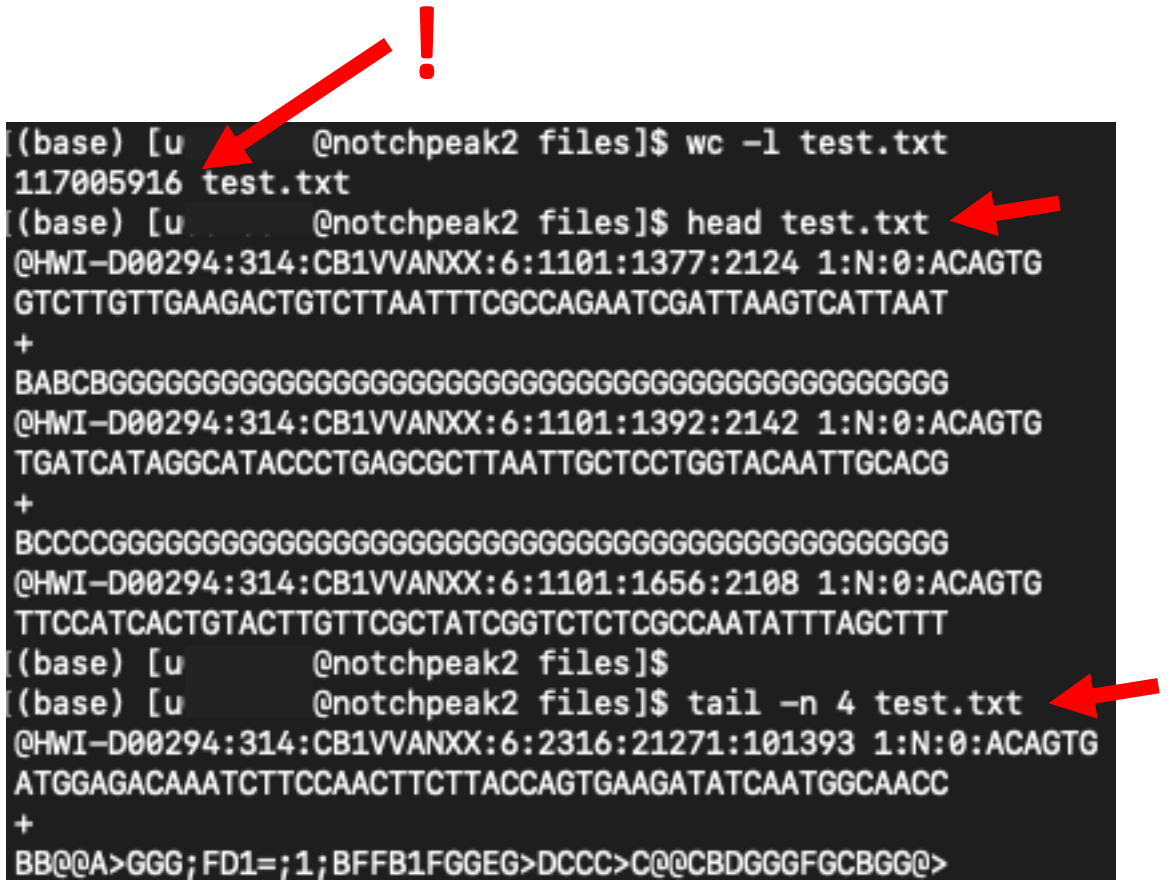
Create one or multiple files, view their contents, redirect file output

```
(base) [u       @notchpeak2 han]$ echo "I will fulfill our destiny" > kylo.txt
(base) [u       @notchpeak2 han]$ cat kylo.txt
I will fulfill our destiny
(base) [u       @notchpeak2 han]$ echo "I will finish what you started" >> kylo.txt
(base) [u       @notchpeak2 han]$ cat kylo.txt
I will fulfill our destiny
I will finish what you started
(base) [u       @notchpeak2 han]$ echo "No one will stand in our way" > kylo.txt
(base) [u       @notchpeak2 han]$ cat kylo.txt
No one will stand in our way
```

How to I read a sample of a file?

- $ head
- $ head -n 15
- $ tail
- $ tail -n 5

- By default, each will read the first or last 10 lines, but can be modified with the -n argument



How do I handle reading a large file?

- $ less -S *filename*

- Use arrow keys to move up/down left/right

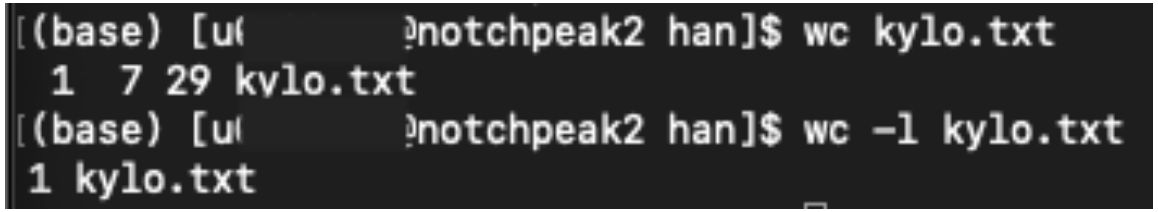
How do I print out metadata about a file?

- Word count

$ wc

Print newline (number of lines),

word, and byte counts for each file



Delete a file or folder

- BE CAREFUL, ONCE YOU DELETE A FILE OR FOLDER IT'S GONE FOREVER!!!

$ rm *filename*

$ rm -r *directoryname*

How do I logout from from the supercomputer?

$ logout

or

$ exit

Review commands

- pwd: print current directory
- cd: move to a new directory
- ls: list files and folders in a directory
- mkdir: make a new directory
- touch: make a new file
- cat: read a file
- head: Read the first n lines
- tail: Read the last n lines
- wc: display the number of lines, words, and characters
- less –S: read and scroll through a file

- . : current directory
- .. : back a directory
- > : Write to file
- >> : Append to file

CAUTION!:
NO TAKEBACKS!

- rm: remove file
- rm –r: remove folder

Homework

Run and copy these commands and outputs to a file

1. Log into the supercomputer
2. Print your current working directory to the console
3. Make a folder called “class_1”
4. Navigate into this new folder
5. Make a new file called “test.txt” within the folder
6. Append “Hello World” to the file
7. Print out the number of lines in the file
8. Navigate back to your home directory
9. Logout