$ Welcome to the Matrix
Unix/Linux I

Computational Biology

Lecture 1

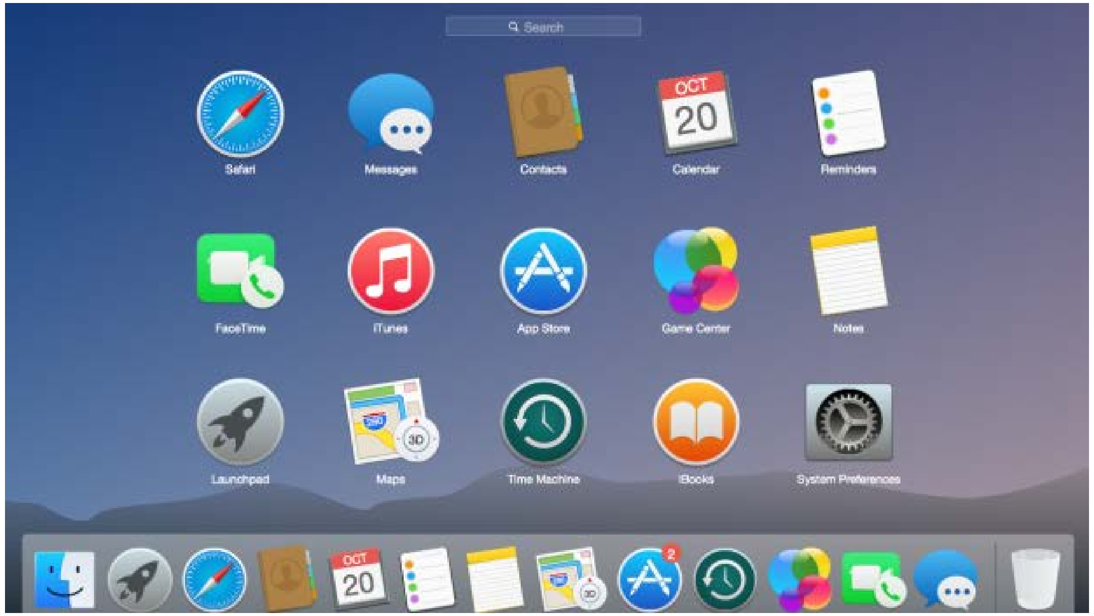
Dr. Chris Bird

LINUX is a Free & Open Source Version of the UNIX Operating System

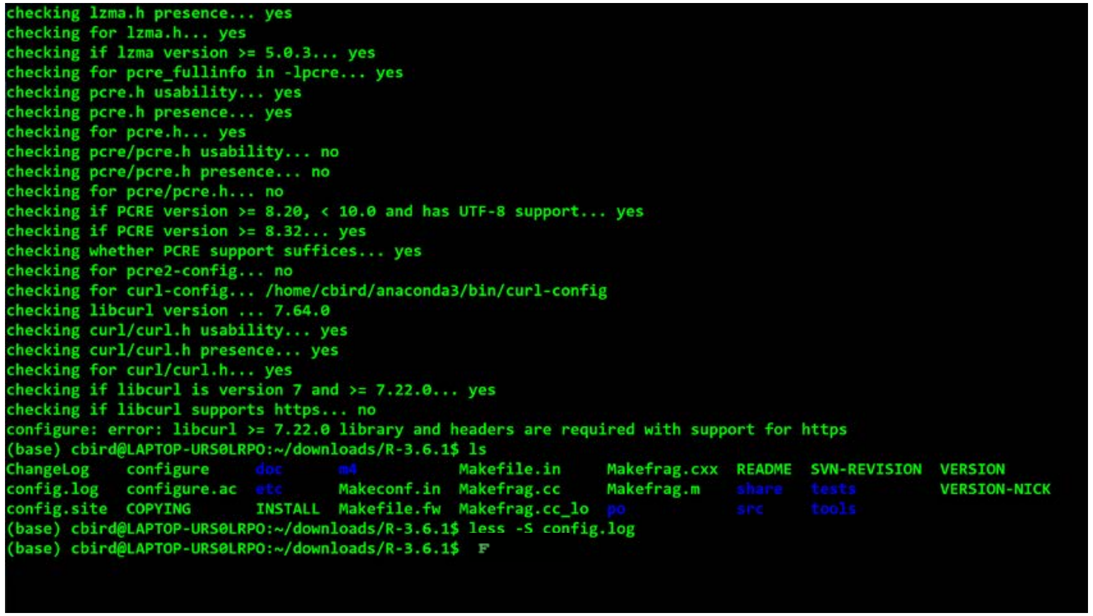
- An operating system is the primary interface between you and the computer
- Open source is a decentralized development model where all aspects of a project are viewable and generally free to use
- Linux is free
  - Supercomputers
  - Useful text manipulation tools

2 Primary Methods of Interfacing with Computers

Graphical User Interface (GUI)



Command-line Interface (CLI)



```
checking lzma.h presence... yes
checking for lzma.h... yes
checking if lzma version >= 5.0.3... yes
checking for pcre_fullinfo in -lpcre... yes
checking pcre.h usability... yes
checking pcre.h presence... yes
checking for pcre.h... yes
checking pcre/pcre.h usability... no
checking pcre/pcre.h presence... no
checking for pcre/pcre.h... no
checking if PCRE version >= 8.20, < 10.0 and has UTF-8 support... yes
checking if PCRE version >= 8.32... yes
checking whether PCRE support suffices... yes
checking for pcre2-config... no
checking for curl-config... /home/cbird/anaconda3/bin/curl-config
checking libcurl version ... 7.64.0
checking curl/curl.h usability... yes
checking curl/curl.h presence... yes
checking for curl/curl.h... yes
checking if libcurl is version 7 and >= 7.22.0... yes
checking if libcurl supports https... no
configure: error: libcurl >= 7.22.0 library and headers are required with support for https
(base) cbird@LAPTOP-URS0LRPO:~/downloads/R-3.6.1$ ls
ChangeLog    configure     doc      m4           Makefile.in     Makefrag.cxx  README  SVN-REVISION  VERSION
config.log   configure.ac  etc      Makeconf.in  Makefrag.cc     Makefrag.m    share   tests         VERSION-NICK
config.site  COPYING       INSTALL  Makefile.fw  Makefrag.cc_lo  po            src     tools
(base) cbird@LAPTOP-URS0LRPO:~/downloads/R-3.6.1$ less -S config.log
(base) cbird@LAPTOP-URS0LRPO:~/downloads/R-3.6.1$  F
```

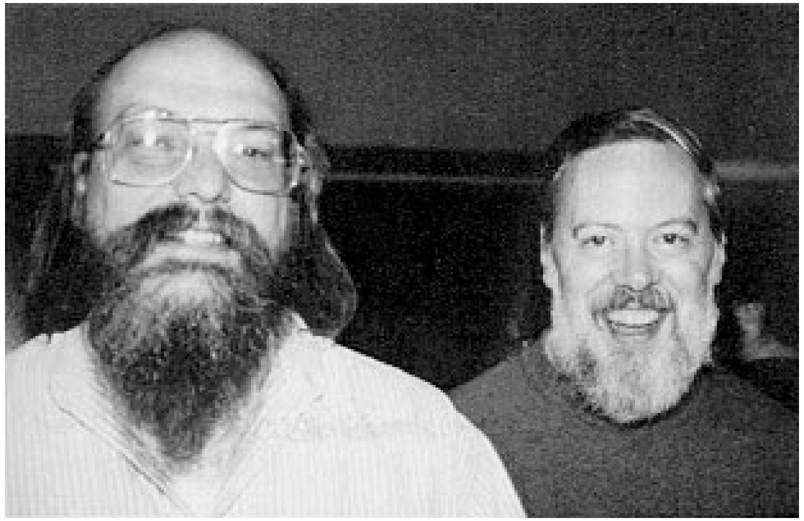
ComputerHope.com

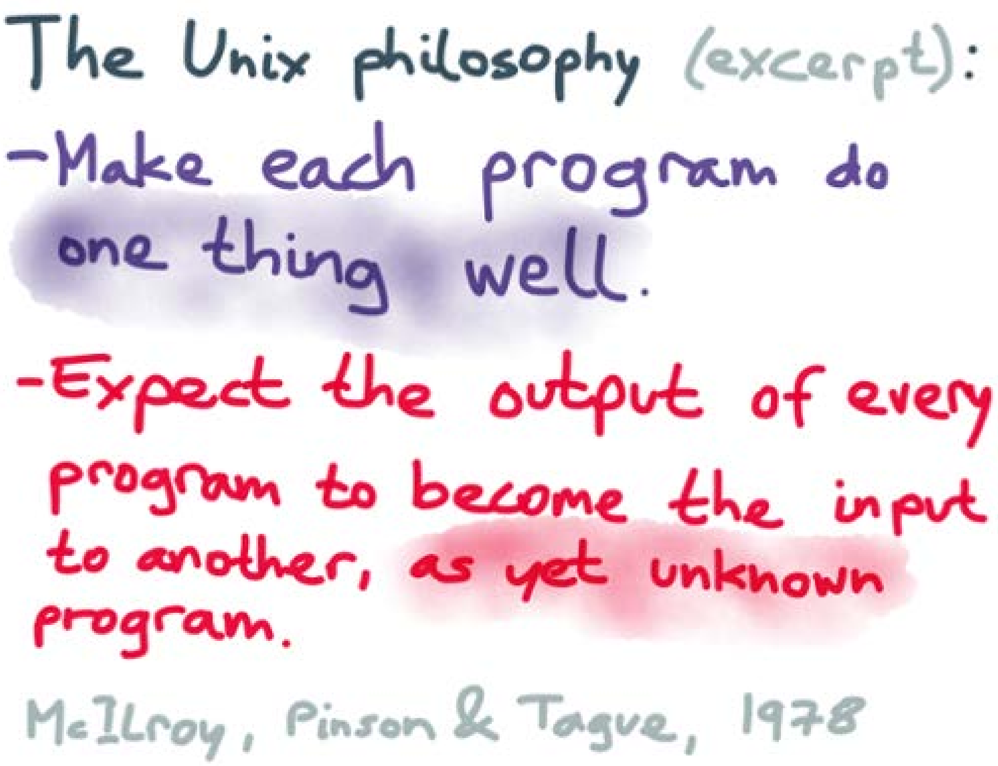
Why use CLI Linux?

- Free
- Automation
- Flexibility
- Powerful
- Designed for developers
- Supercomputers use it
- Many software tools for biologists
- Large body of support online

The UNIX Philosophy

- One program (command) does one thing
- All programs accept input as a text stream and output a modified text stream
- Programs can be linked together into serial pipelines to achieve complex results





The Unix philosophy (excerpt):

-Make each program do one thing well.

-Expect the output of every program to become the input to another, as yet unknown program.

McIlroy, Pinson & Tague, 1978

Linux CLI Pipelines Facilitate Scientific Reproducibility and Long-Term Efficiency

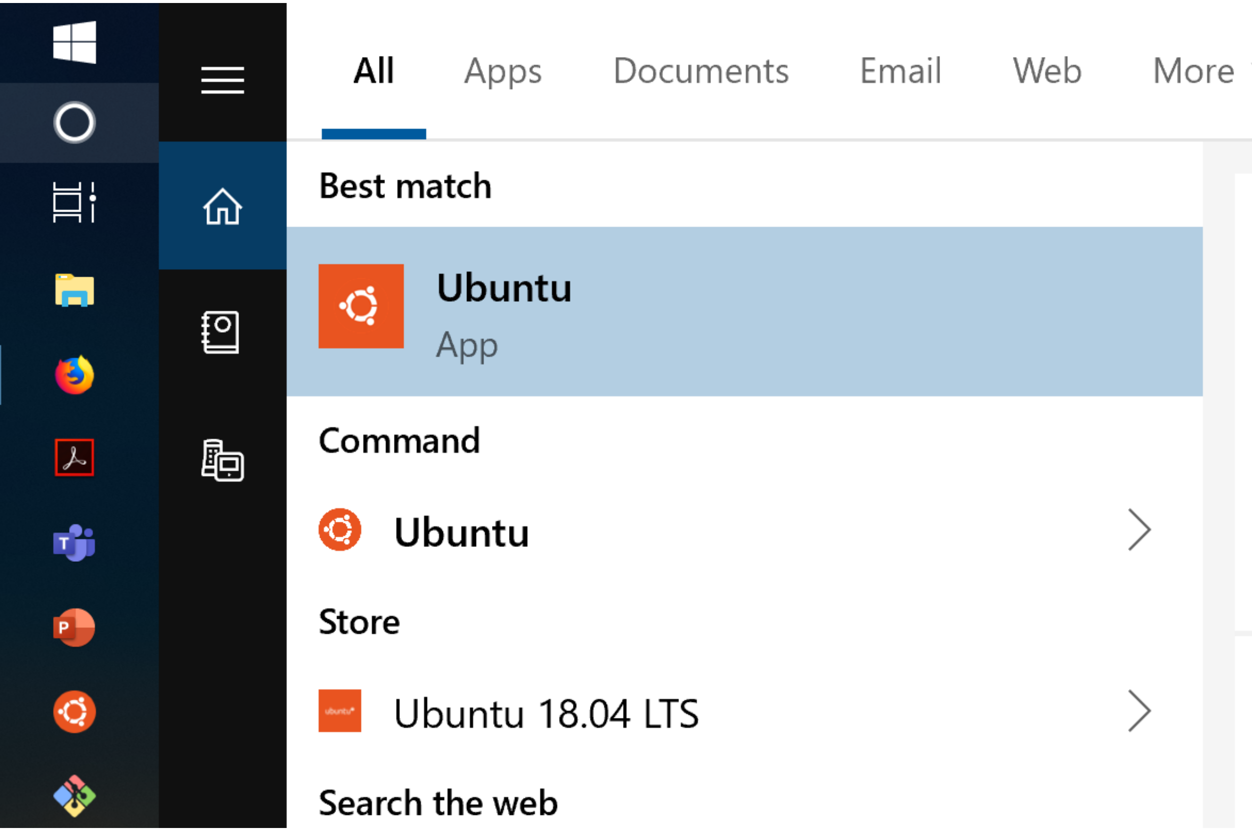
Comparison of GUI and CLI for manipulating data

	GUI	CLI
Learning curve	**Short, shallow**	Long, steep
Amount of your time taken to process large amounts of data	Long	**Short**
Process Documented or Recorded	Often not, mouse clicks	**Always**
Ability to identify mistake	Poor	**Excellent**
Time to recover from mistake	Long	**Short**
Ease for another lab to reproduce	Difficult to impossible	**Simple**

Open A Terminal Window

WIN10: Search Ubuntu

MacOS: Search Terminal



Best match

Ubuntu
App

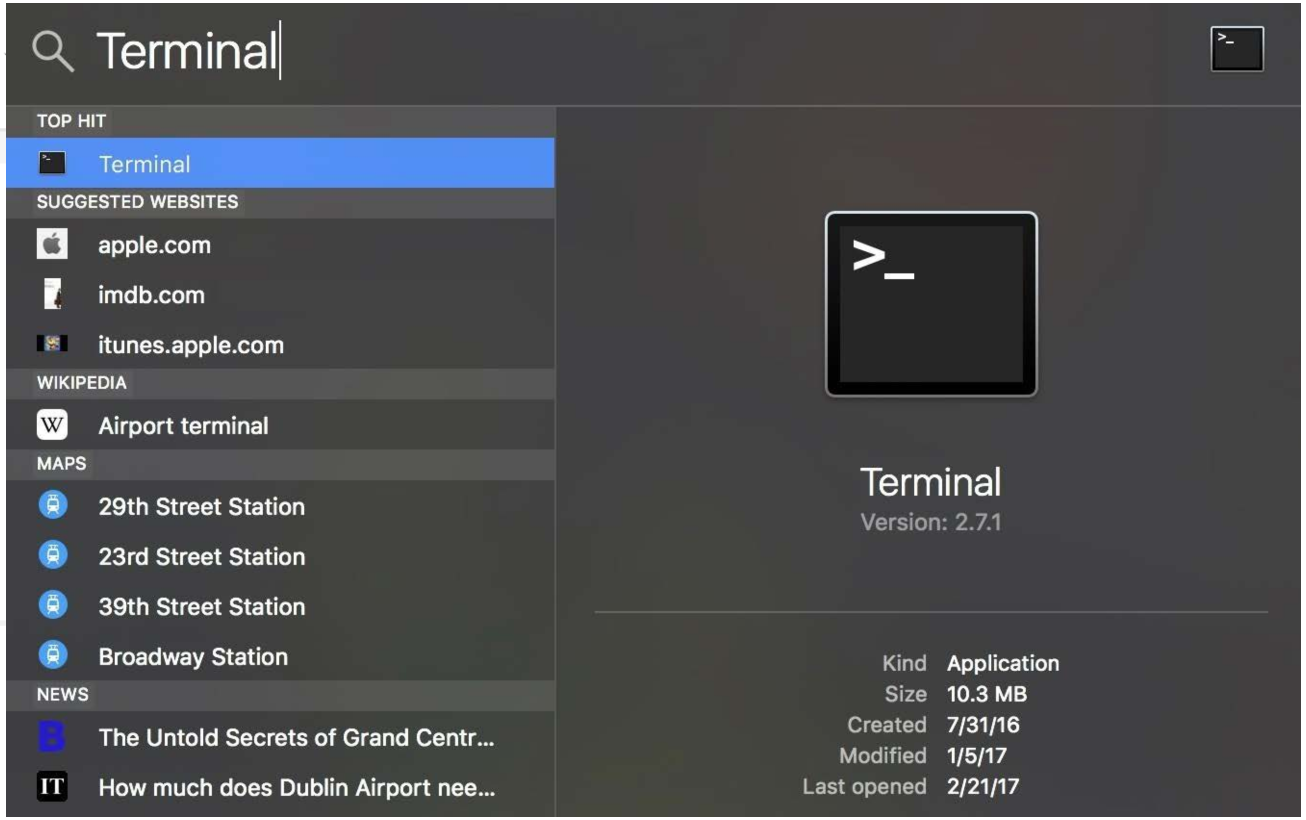
Command

Ubuntu

Store

Ubuntu 18.04 LTS

Search the web



Terminal

TOP HIT

Terminal

SUGGESTED WEBSITES

apple.com

imdb.com

itunes.apple.com

WIKIPEDIA

Airport terminal

MAPS

29th Street Station

23rd Street Station

39th Street Station

Broadway Station

NEWS

The Untold Secrets of Grand Centr...

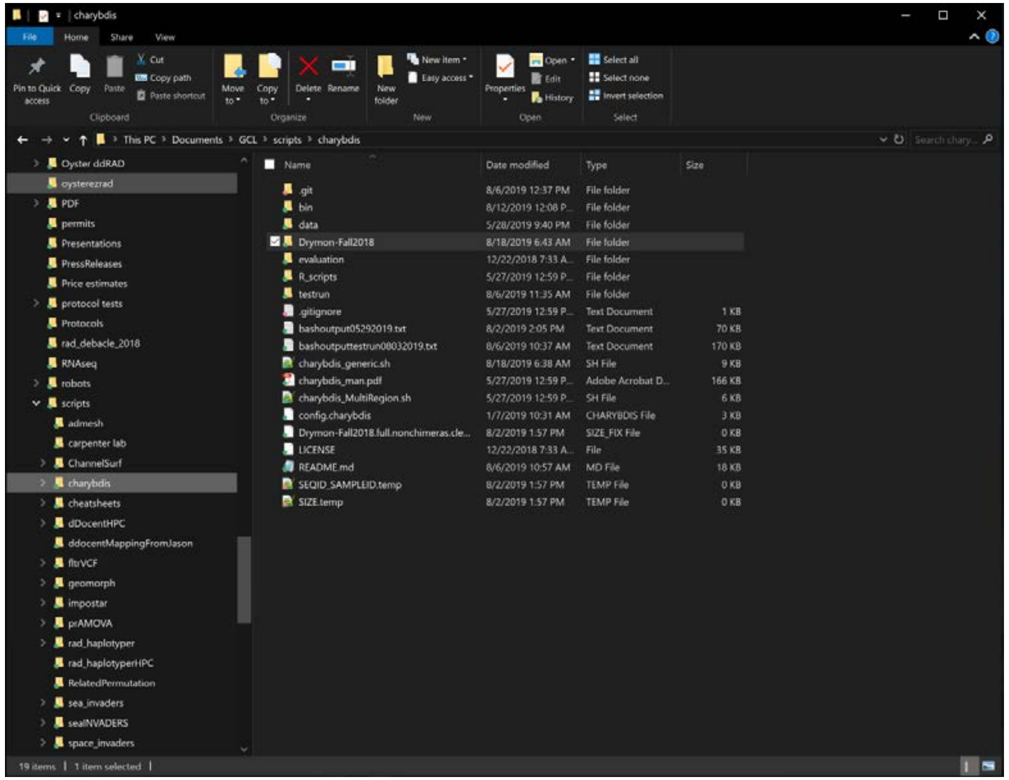
How much does Dublin Airport nee...

Terminal
Version: 2.7.1

Kind	Application
Size	10.3 MB
Created	7/31/16
Modified	1/5/17
Last opened	2/21/17

The Directory Structure is the Organization of Files and Folders (aka Directories) In Your Computer

WIN10 File Explorer



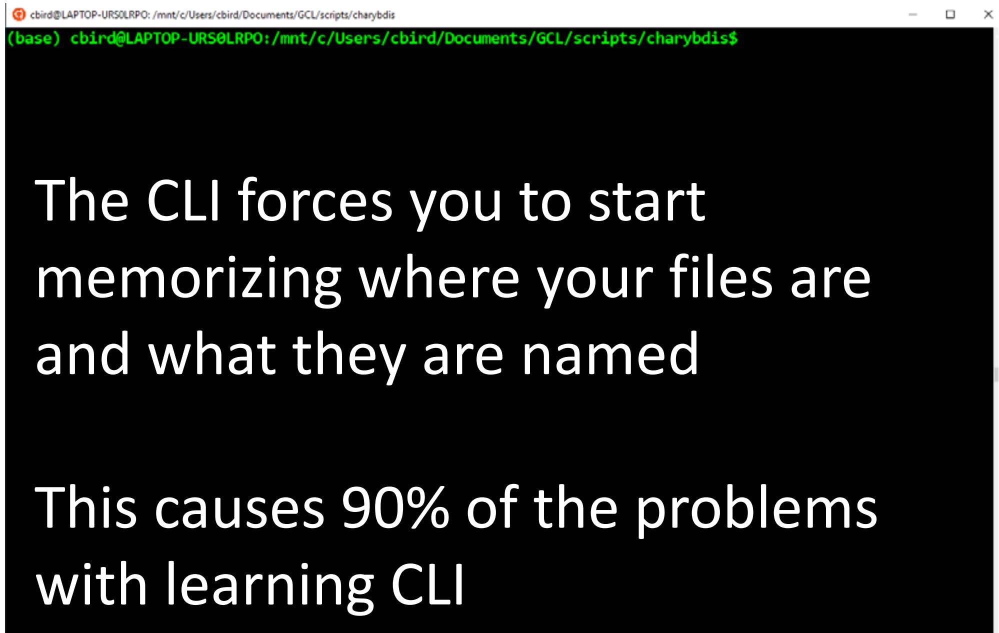
This PC › Documents › GCL › scripts › charybdis

- Oyster ddRAD
- oysterezrad
- PDF
- permits
- Presentations
- PressReleases
- Price estimates
- protocol tests
- Protocols
- rad_debacle_2018
- RNAseq
- robots
- scripts
  - admesh
  - carpenter lab
  - ChannelSurf
  - charybdis
  - cheatsheets
  - dDocentHPC
  - ddocentMappingFromJason
  - fltrVCF
  - geomorph
  - impostar
  - prAMOVA
  - rad_haplotyper
  - rad_haplotyperHPC
  - RelatedPermutation
  - sea_invaders
  - seaINVADERS
  - space_invaders

Name	Date modified	Type	Size
.git	8/6/2019 12:37 PM	File folder	
bin	8/12/2019 12:08 P...	File folder	
data	5/28/2019 9:40 PM	File folder	
☑ Drymon-Fall2018	8/18/2019 6:43 AM	File folder	
evaluation	12/22/2018 7:33 A...	File folder	
R_scripts	5/27/2019 12:59 P...	File folder	
testrun	8/6/2019 11:35 AM	File folder	
.gitignore	5/27/2019 12:59 P...	Text Document	1 KB
bashoutput05292019.txt	8/2/2019 2:05 PM	Text Document	70 KB
bashoutputtestrun08032019.txt	8/6/2019 10:37 AM	Text Document	170 KB
charybdis_generic.sh	8/18/2019 6:38 AM	SH File	9 KB
charybdis_man.pdf	5/27/2019 12:59 P...	Adobe Acrobat D...	166 KB
charybdis_MultiRegion.sh	5/27/2019 12:59 P...	SH File	6 KB
config.charybdis	1/7/2019 10:31 AM	CHARYBDIS File	3 KB
Drymon-Fall2018.full.nonchimeras.cle...	8/2/2019 1:57 PM	SIZE_FIX File	0 KB
LICENSE	12/22/2018 7:33 A...	File	35 KB
README.md	8/6/2019 10:57 AM	MD File	18 KB
SEQID_SAMPLEID.temp	8/2/2019 1:57 PM	TEMP File	0 KB
SIZE.temp	8/2/2019 1:57 PM	TEMP File	0 KB

19 items | 1 item selected |

Ubuntu Terminal

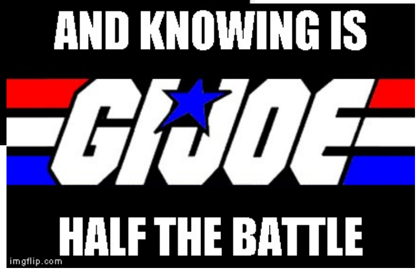


cbird@LAPTOP-URS0LRPO: /mnt/c/Users/cbird/Documents/GCL/scripts/charybdis

```
(base) cbird@LAPTOP-URS0LRPO:/mnt/c/Users/cbird/Documents/GCL/scripts/charybdis$
```

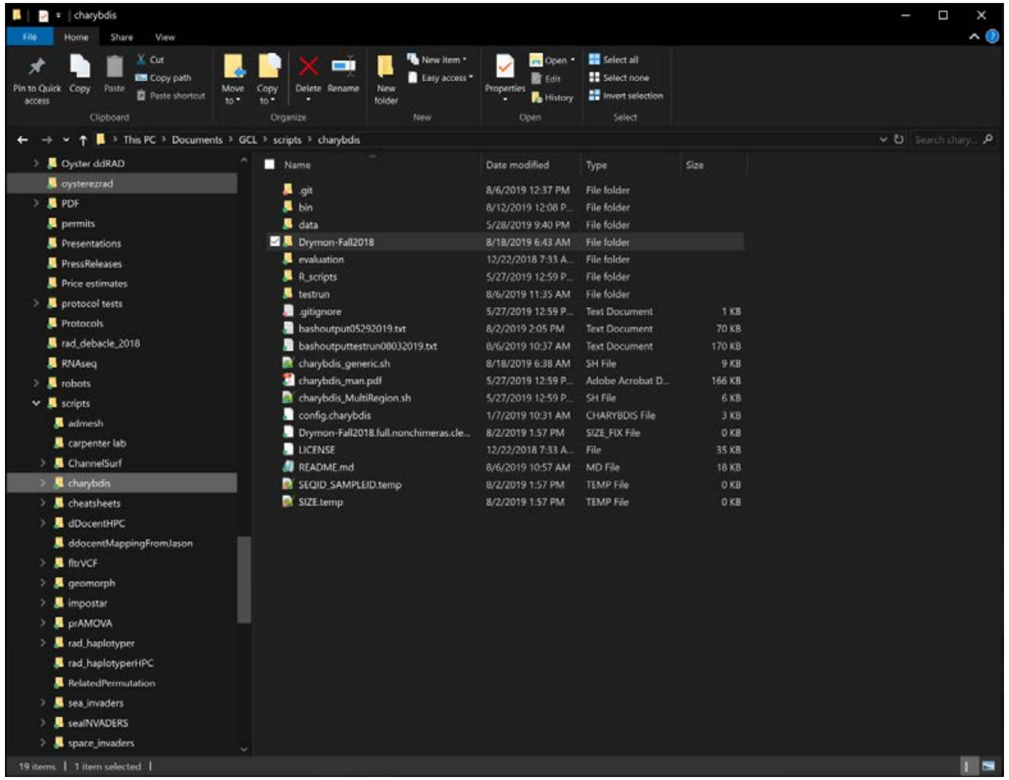
The CLI forces you to start memorizing where your files are and what they are named

This causes 90% of the problems with learning CLI



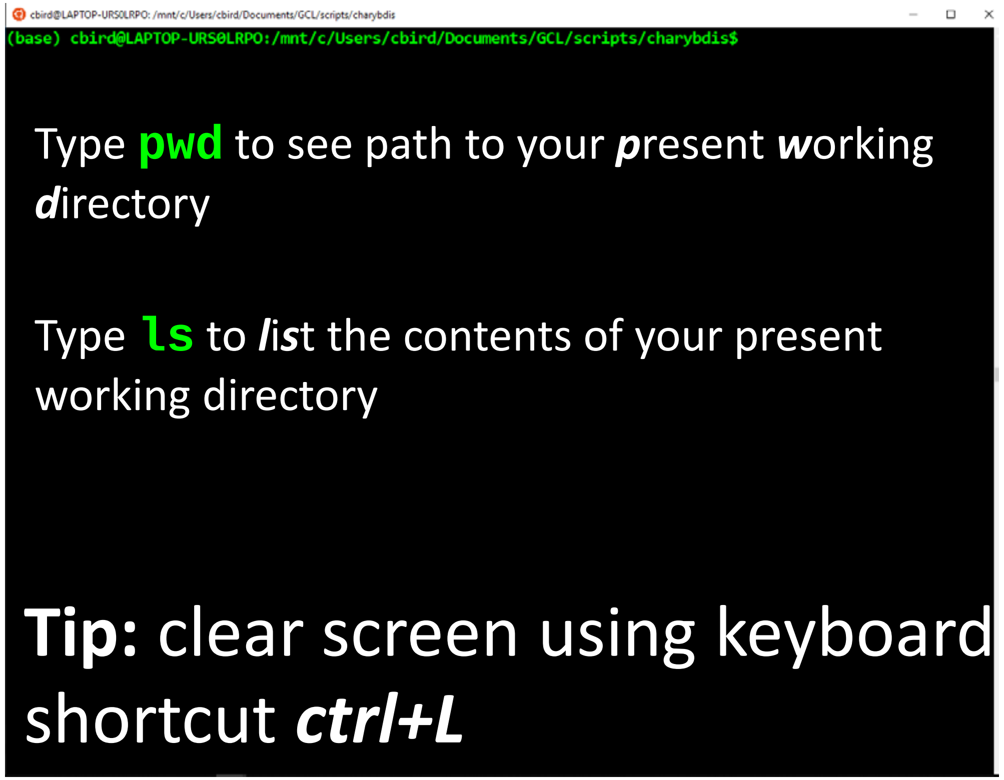
The Directory Structure is the Organization of Files and Folders (aka Directories) In Your Computer

WIN10 File Explorer



Name	Date modified	Type	Size
.git	8/6/2019 12:37 PM	File folder	
bin	8/12/2019 12:08 P...	File folder	
data	5/28/2019 9:40 PM	File folder	
☑ Drymon-Fall2018	8/18/2019 6:43 AM	File folder	
evaluation	12/22/2018 7:33 A...	File folder	
R_scripts	5/27/2019 12:59 P...	File folder	
testrun	8/6/2019 11:35 AM	File folder	
.gitignore	5/27/2019 12:59 P...	Text Document	1 KB
bashoutput05292019.txt	8/2/2019 2:05 PM	Text Document	70 KB
bashoutputtestrun08032019.txt	8/6/2019 10:37 AM	Text Document	170 KB
charybdis_generic.sh	8/18/2019 6:38 AM	SH File	9 KB
charybdis_man.pdf	5/27/2019 12:59 P...	Adobe Acrobat D...	166 KB
charybdis_MultiRegion.sh	5/27/2019 12:59 P...	SH File	6 KB
config.charybdis	1/7/2019 10:31 AM	CHARYBDIS File	3 KB
Drymon-Fall2018.full.nonchimeras.cle...	8/2/2019 1:57 PM	SIZE_FIX File	0 KB
LICENSE	12/22/2018 7:33 A...	File	35 KB
README.md	8/6/2019 10:57 AM	MD File	18 KB
SEQID_SAMPLEID.temp	8/2/2019 1:57 PM	TEMP File	0 KB
SIZE.temp	8/2/2019 1:57 PM	TEMP File	0 KB

Ubuntu Terminal


cbird@LAPTOP-URS0LRPO: /mnt/c/Users/cbird/Documents/GCL/scripts/charybdis

```
(base) cbird@LAPTOP-URS0LRPO:/mnt/c/Users/cbird/Documents/GCL/scripts/charybdis$
```

Type **pwd** to see path to your ***p****resent* ***w****orking* ***d****irectory*

Type **ls** to *li*s*t* the contents of your present working directory

Tip: clear screen using keyboard shortcut ***ctrl+L***

Unix/Linux Command Line Terminology

The **path** is the address of a file or directory in the directory structure

Description	*Path* in Unix, Linux, Ubuntu, MacOS, Android	Path in Windows
Root, or top of the directory tree	/	c:\
A **file** named file.txt in the root dir	/file.txt	c:\file.txt
A **directory** named folder1 in the root dir	/folder1	c:\folder1
A file named dna.txt in folder1	/folder1/dna.txt	c:\folder1\dna.txt

Important Directories

`/bin`

- Contains several basic programs

`/dev`

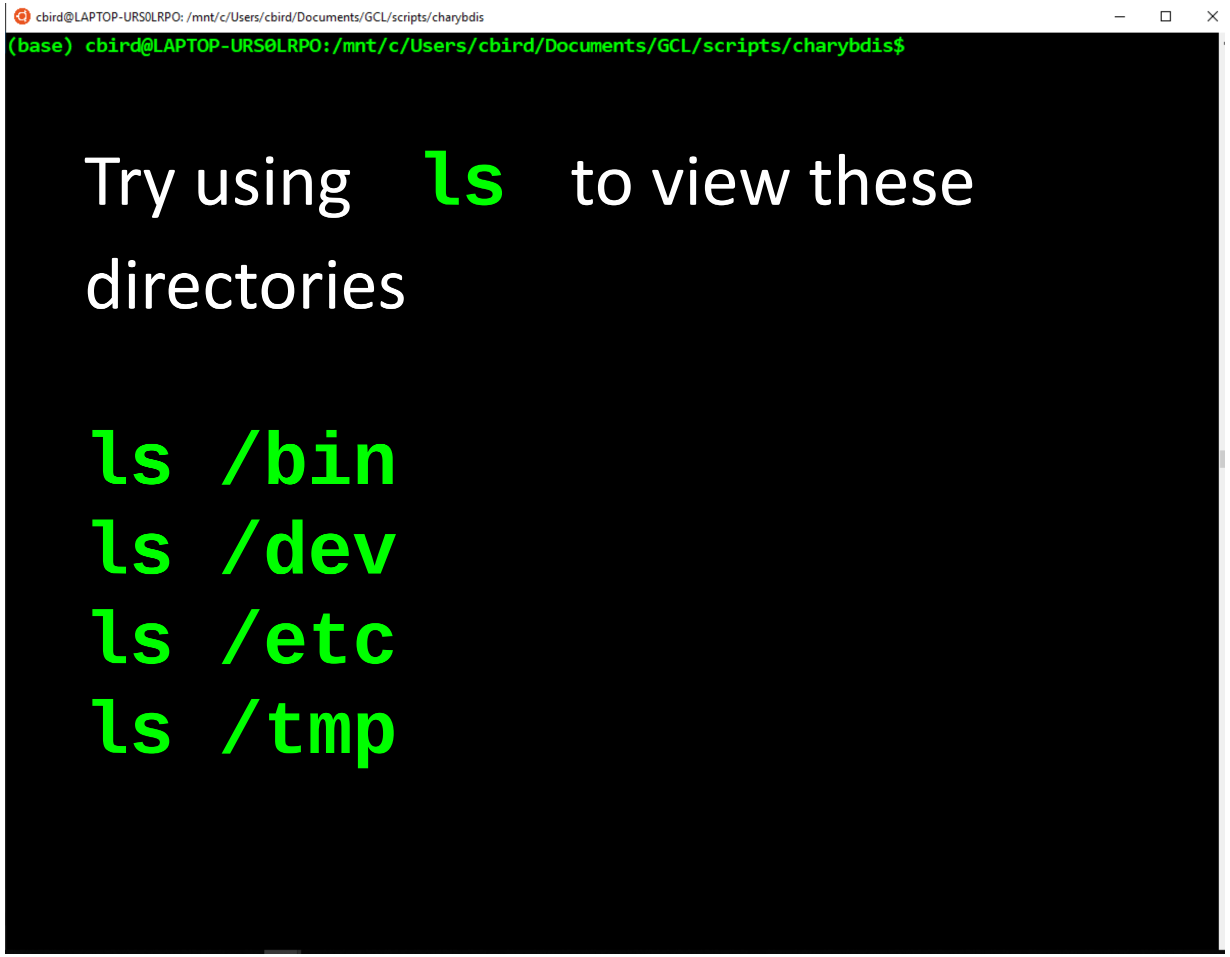
- Contains the files connecting to devices such as the keyboard, mouse, and screen

`/etc`

- Contains configuration files

`/tmp`

- Contains temporary files



cbird@LAPTOP-URS0LRPO: /mnt/c/Users/cbird/Documents/GCL/scripts/charybdis

```
(base) cbird@LAPTOP-URS0LRPO:/mnt/c/Users/cbird/Documents/GCL/scripts/charybdis$
```

Try using `ls` to view these directories

```
ls /bin
ls /dev
ls /etc
ls /tmp
```

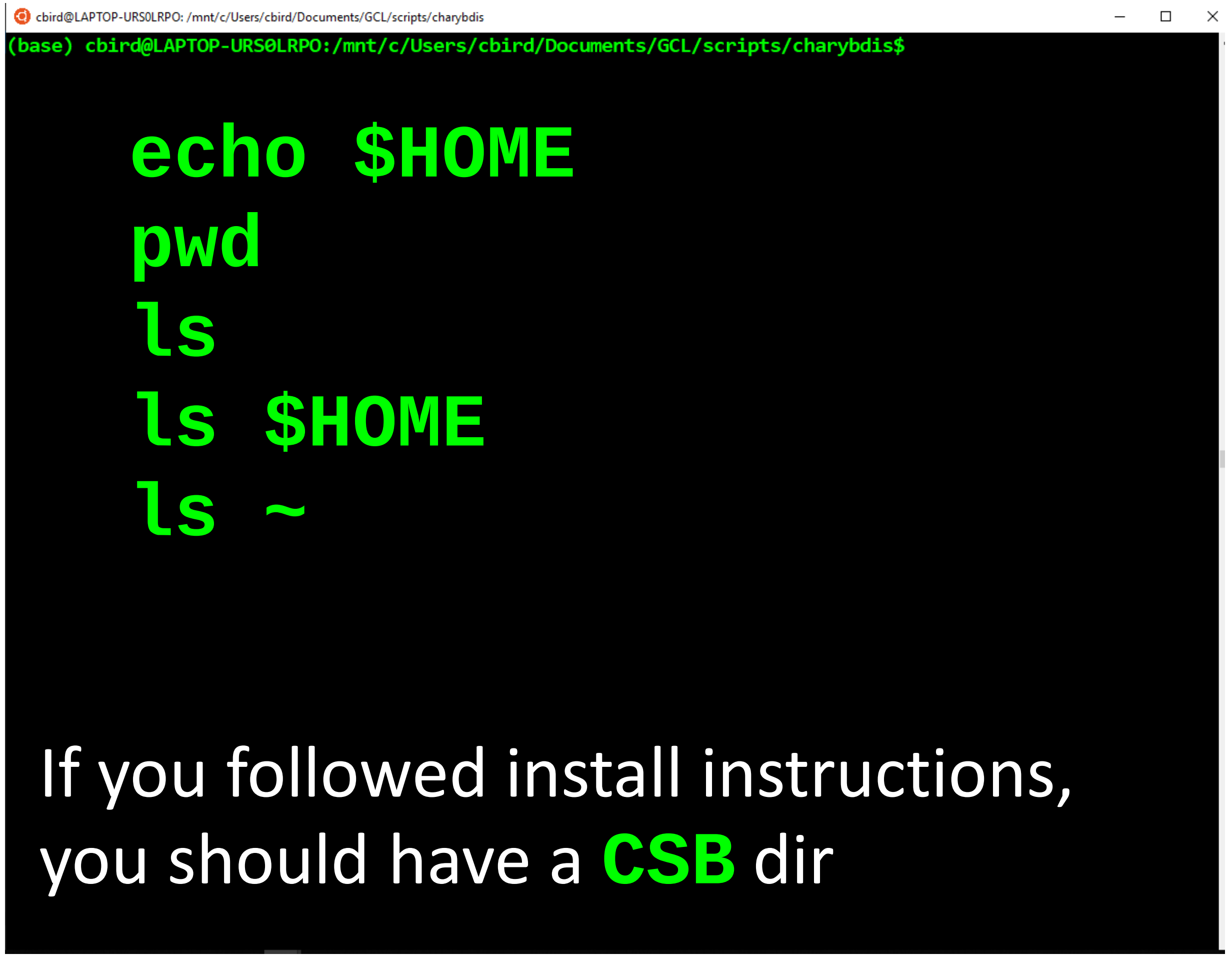
Your Home Directory

/username/home

- Starting or login directory
- Specific to user
- Place for personal files, dirs, programs, downloads etc

$HOME

- The **path** to your home dir is stored in this **variable**
- A variable stores information
- Always preceded by a **$** after it is created
- **$HOME** is an **environmental variable** created by the operating system and bash

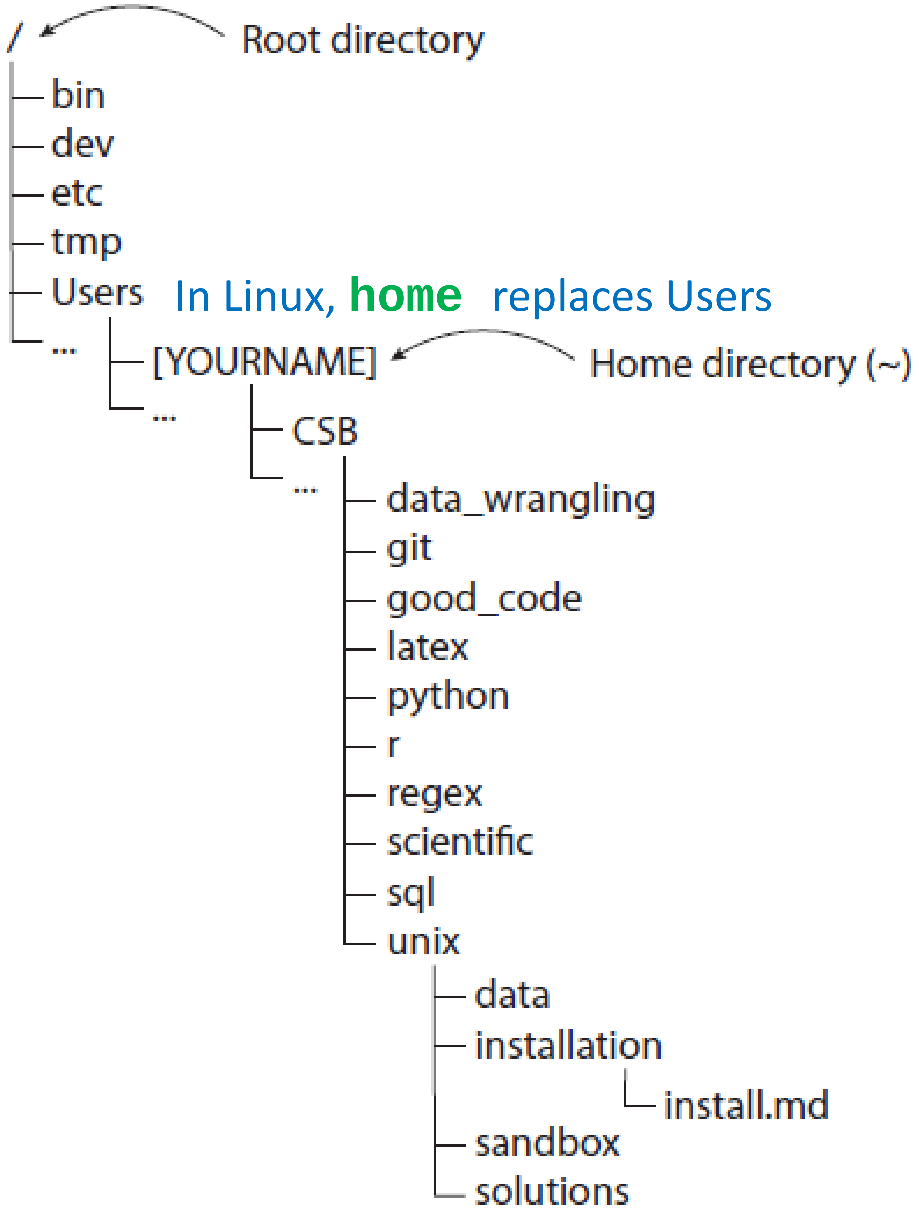

cbird@LAPTOP-URS0LRPO: /mnt/c/Users/cbird/Documents/GCL/scripts/charybdis

```
(base) cbird@LAPTOP-URS0LRPO:/mnt/c/Users/cbird/Documents/GCL/scripts/charybdis$

    echo $HOME
    pwd
    ls
    ls $HOME
    ls ~
```

If you followed install instructions, you should have a **CSB** dir

Directory Tree

Showing Contents of $HOME/CSB/unix/installation



/ Root directory

bin

dev

etc

tmp

Users In Linux, **home** replaces Users

...

[YOURNAME] Home directory (~)

...

CSB

...

data_wrangling

git

good_code

latex

python

r

regex

scientific

sql

unix

data

installation

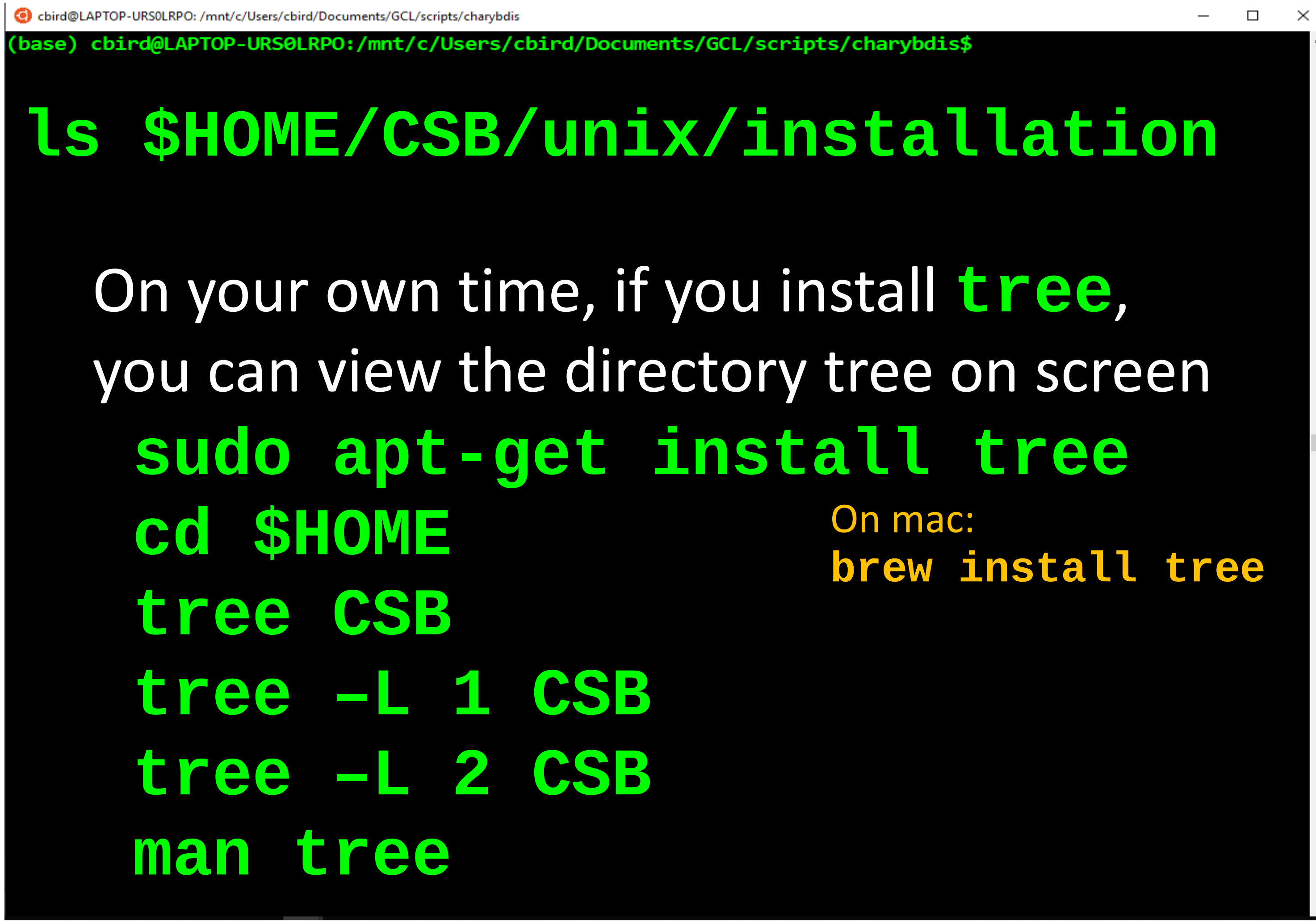
install.md

sandbox

solutions

Full path of the file install.md:

`/Users/[YOURNAME]/CSB/unix/installation/install.md`



cbird@LAPTOP-URS0LRPO: /mnt/c/Users/cbird/Documents/GCL/scripts/charybdis

```
(base) cbird@LAPTOP-URS0LRPO:/mnt/c/Users/cbird/Documents/GCL/scripts/charybdis$

ls $HOME/CSB/unix/installation
```

On your own time, if you install **tree**, you can view the directory tree on screen

```
sudo apt-get install tree
cd $HOME
tree CSB
tree –L 1 CSB
tree –L 2 CSB
man tree
```

On mac:

```
brew install tree
```

CSB/unix Repository

CSB/unix/data

- Contains data for examples and exercises

CSB/unix/installation

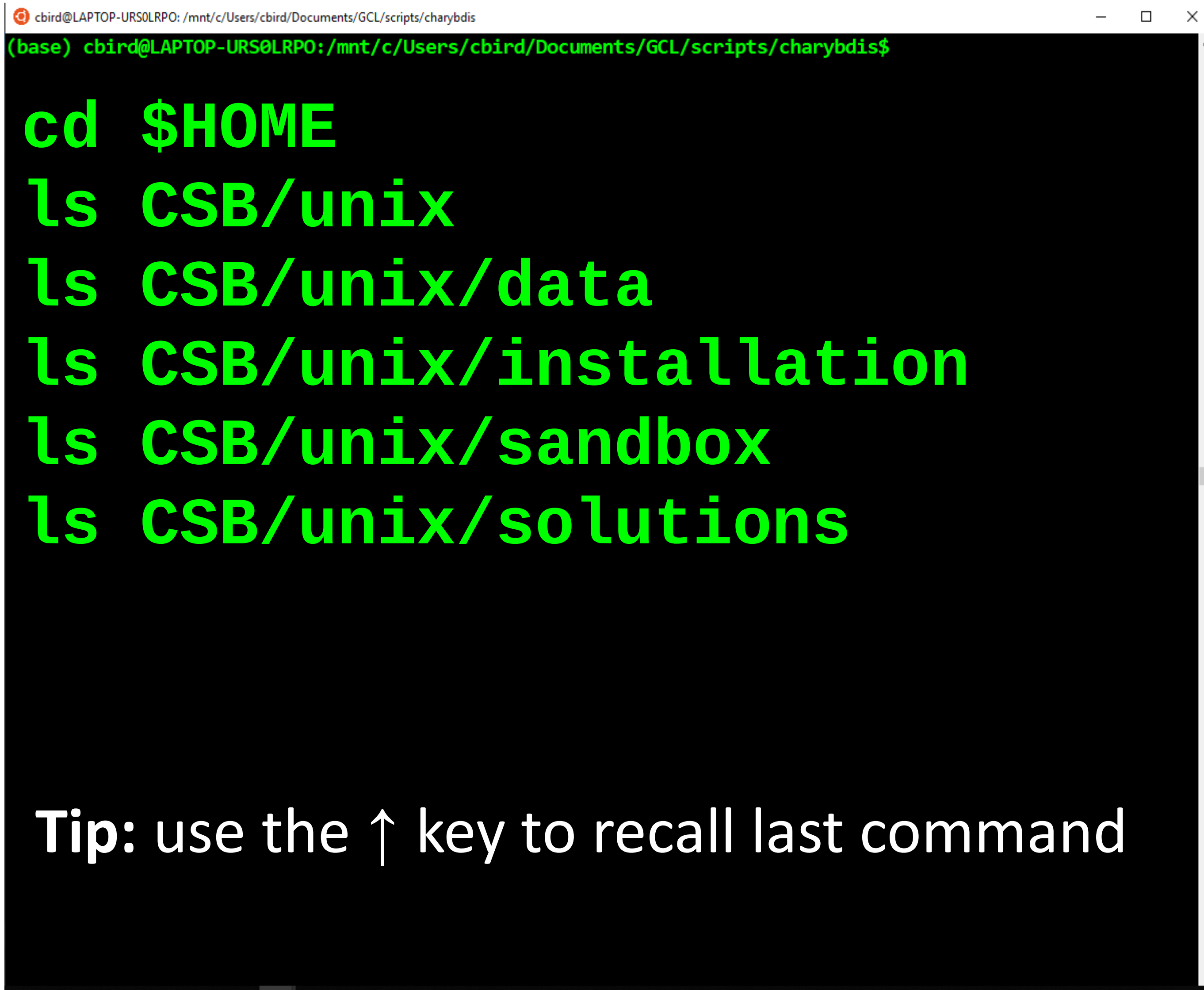
- Contains instructions for installing software for this chapter

CSB/unix/sandbox

- Dir where we work and experiment

CSB/unix/solutions

- Solutions in code (bash) pseudocode (plain English) for your consultation when you get stuck with an exercise.


cbird@LAPTOP-URSOLRPO: /mnt/c/Users/cbird/Documents/GCL/scripts/charybdis

```
(base) cbird@LAPTOP-URS0LRPO:/mnt/c/Users/cbird/Documents/GCL/scripts/charybdis$

cd $HOME
ls CSB/unix
ls CSB/unix/data
ls CSB/unix/installation
ls CSB/unix/sandbox
ls CSB/unix/solutions
```

Tip: use the ↑ key to recall last command

$ Welcome to the Matrix
1.4 The Shell

Computational Biology

Lecture 1

Dr. Chris Bird

The Shell

- The **shell** is software that controls the operating system **kernel** and is accessed through a **terminal** window
- The shell we are using in Ubuntu and MacOS is **BASH**, or **Born Again Shell**
- The **commands** we've been using are BASH commands which allow us to control the operating system

~

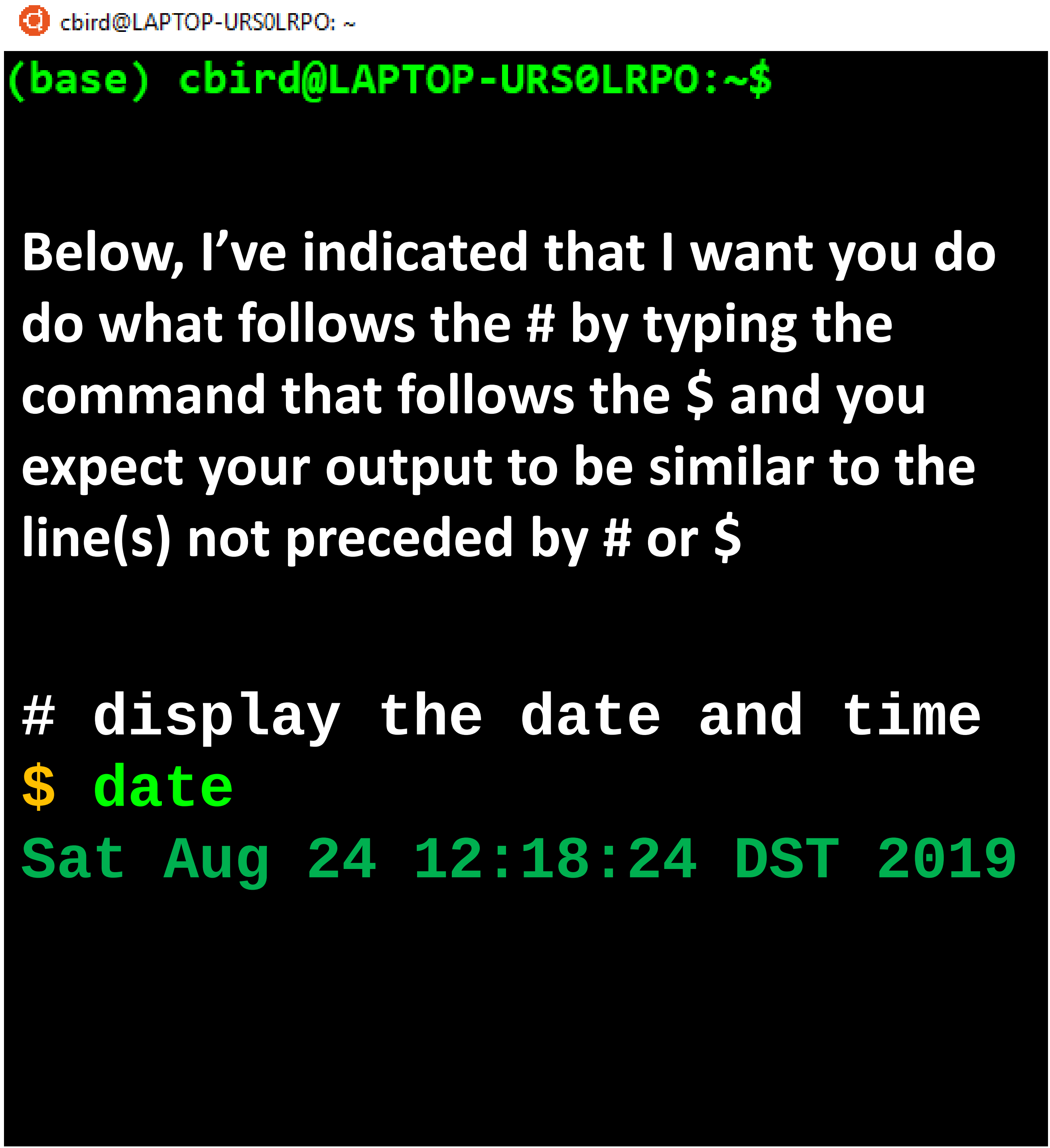
- Indicates where I am, the home dir

$

- Indicates the terminal is ready to accept commands
- From here forward, $<space><command> indicates you should type the command into the terminal

#

- A hash symbol means that everything that follows is a comment, usually in Engish



```
(base) cbird@LAPTOP-URS0LRPO:~$
```

Below, I've indicated that I want you do do what follows the # by typing the command that follows the $ and you expect your output to be similar to the line(s) not preceded by # or $

```
# display the date and time
$ date
Sat Aug 24 12:18:24 DST 2019
```

Bash Keyboard Shortcuts

↑	Scroll through previous commands
Tab	autocomplete command, dir, or file name if you hit tab and nothing happens there's either multiple matches or 0 matches
Tab,Tab	show matches
Ctrl+A	Go to the beginning of the line.
Ctrl+E	Go to the end of the line.
Ctrl+L	Clear the screen.
Ctrl+U	Clear the line before the cursor position.
Ctrl+K	Clear the line after the cursor.
Ctrl+C	Kill the command that is currently running.
Ctrl+D	Exit the current shell.
Alt+F	Move cursor forward one word (in OS X, Esc+F).
Alt+B	Move cursor backward one word (in OS X, Esc+B).

```
(base) cbird@LAPTOP-URS0LRPO:~$

# try some of the shortcuts
$
```

Bash Commands

`cal 2020 -j`

- **Commands** like `cal` are programs that follow the UNIX philosophy
- **Arguments** like `2020` are essentially options, order usually matters and some commands require particular arguments
  - **cp** or copy requires at least which file to copy and where to copy it, in that order
- `-j` is an **option**, in this case it means Julian calendar
  - --julian is the same as –j, options that are words are always preceded by two dashes

```
# print calendar
$ cal
 August 2019
Su Mo Tu We Th Fr Sa
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

$ cal 2020
$ cal –j
$ cal --julian
$ cal –j 2020
```

If you want to stop a command, ***ctrl+c***

Getting Bash Help

- It’s impossible to remember all command and arguments

- If you know what you want to do, but you don’t know the command
  - Google search “bash <English description of what you want to do>”

- If you know the command, but you don’t know the arguments
  - `man <CommandName>`
  - All manuals have same format

```
# view calendar manual
$ man cal
NAME
  <name and brief descrip>
SYNOPSIS
  <examples of how to run>
DESCRIPTION
  <detailed description>
  <list of arguments/options>
```

Tip: scroll with arrow keys and close manual with ***q*** key

Changing Directories

`cd ..`

- Move up to parent directory

`cd /`

- Move to root directory

`cd ~`

- Move to home directory

`cd -`

- Move to last directory

`pwd`

- Path to present working dir

`ls`

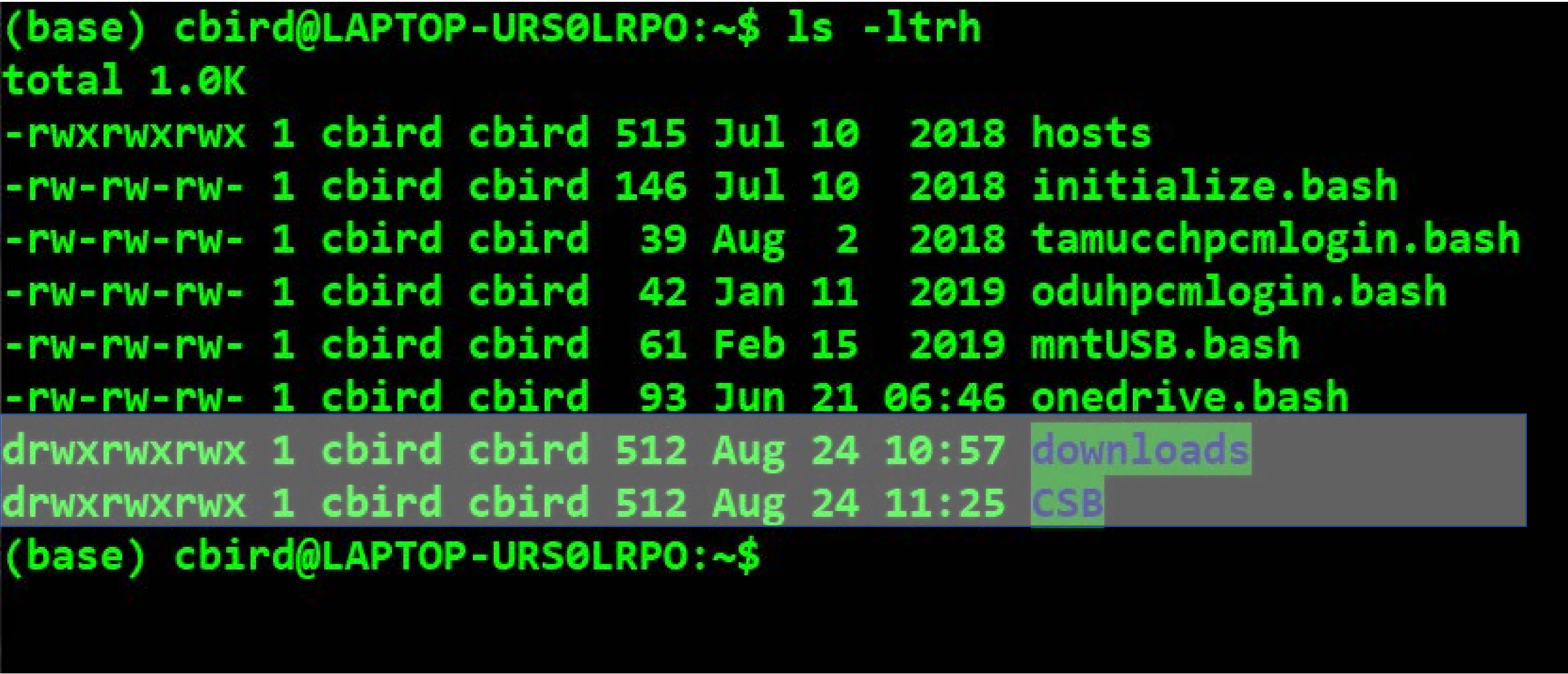
- Show contents of present directory

```
# move around dir system
$ cd ..
$ pwd
$ cd /
$ pwd
$ cd -
$ pwd
$ cd ~
$ pwd
# show dir contents
$ ls
$ ls -l
$ ls -ltrh
```

Note: single letter *options* can typically be combined together, `-l -t -r -h`

Interpreting Output of `ls -l`

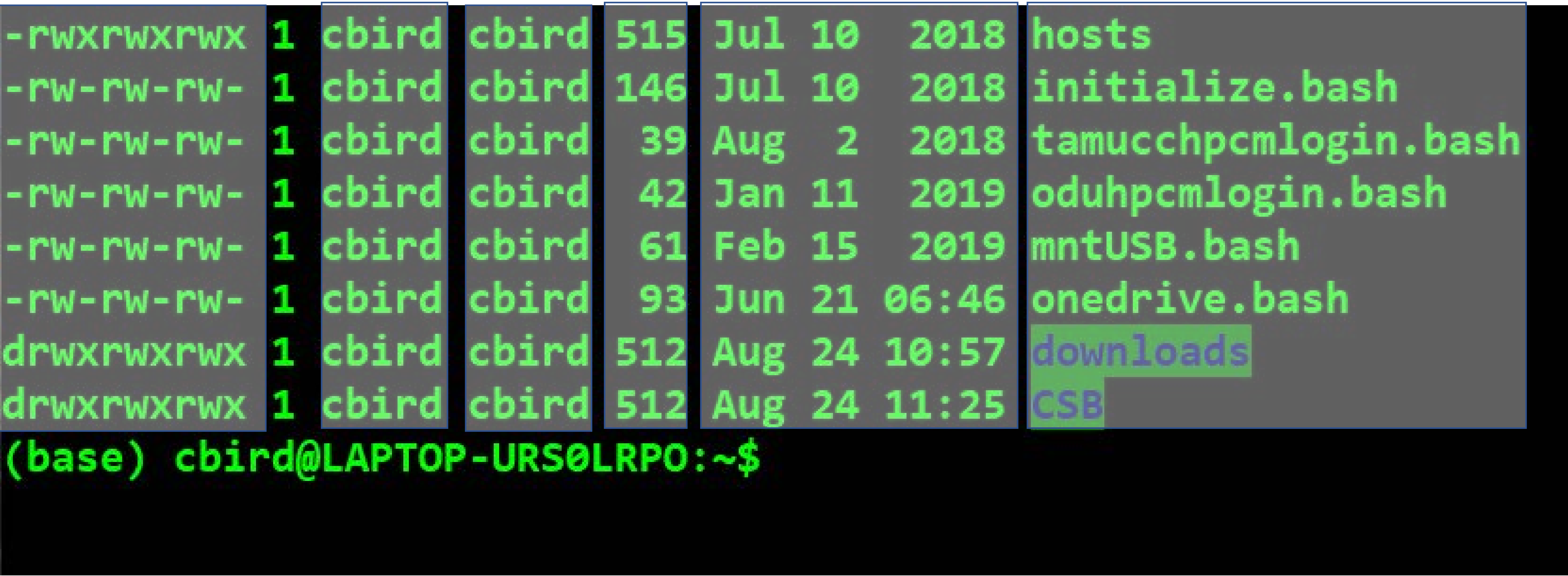
Dirs are highlighted below, files are not



```
(base) cbird@LAPTOP-URS0LRPO:~$ ls -ltrh
total 1.0K
-rwxrwxrwx 1 cbird cbird 515 Jul 10  2018 hosts
-rw-rw-rw- 1 cbird cbird 146 Jul 10  2018 initialize.bash
-rw-rw-rw- 1 cbird cbird  39 Aug  2  2018 tamucchpcmlogin.bash
-rw-rw-rw- 1 cbird cbird  42 Jan 11  2019 oduhpcmlogin.bash
-rw-rw-rw- 1 cbird cbird  61 Feb 15  2019 mntUSB.bash
-rw-rw-rw- 1 cbird cbird  93 Jun 21 06:46 onedrive.bash
drwxrwxrwx 1 cbird cbird 512 Aug 24 10:57 downloads
drwxrwxrwx 1 cbird cbird 512 Aug 24 11:25 CSB
(base) cbird@LAPTOP-URS0LRPO:~$
```

Interpreting Output of `ls -l`

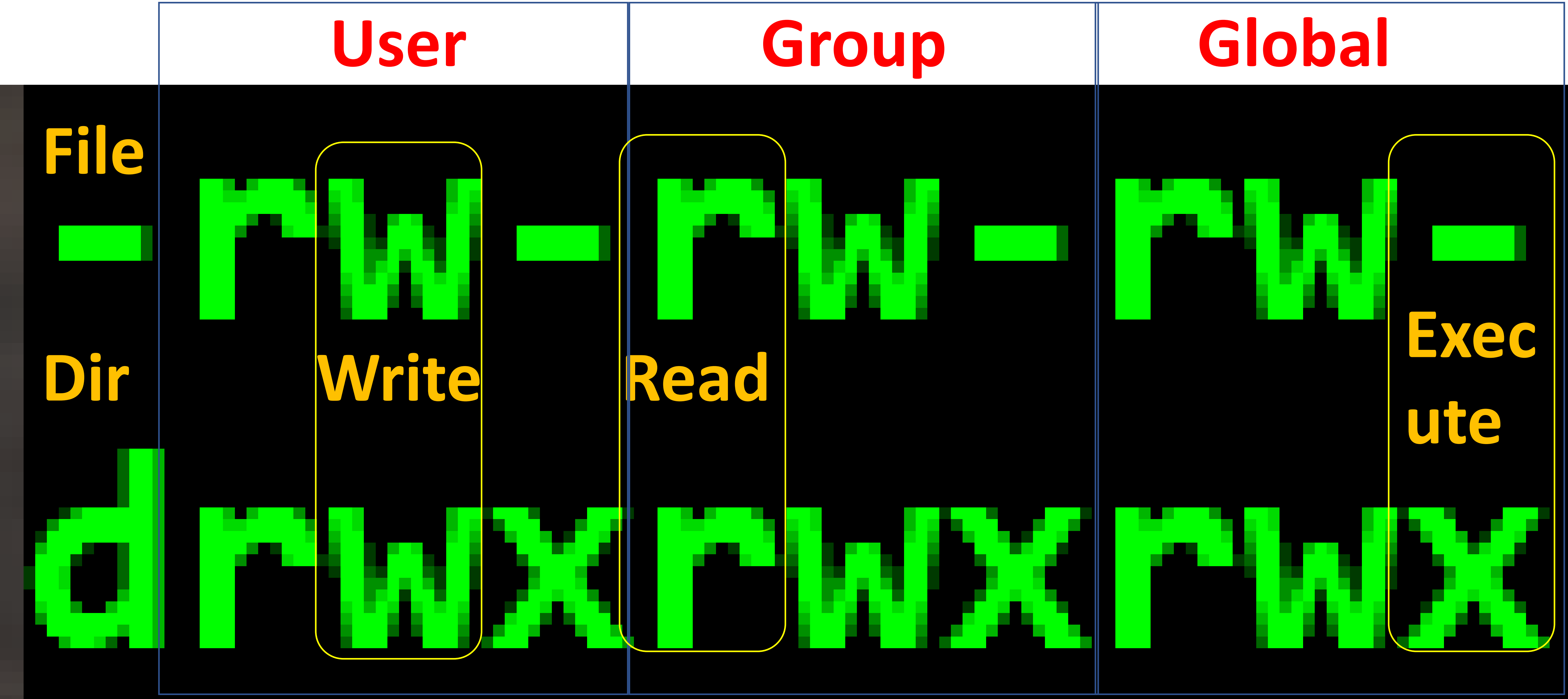
Next Slide   Usr   Grp   Size   Date   Names



```
-rwxrwxrwx 1 cbird cbird 515 Jul 10  2018 hosts
-rw-rw-rw- 1 cbird cbird 146 Jul 10  2018 initialize.bash
-rw-rw-rw- 1 cbird cbird  39 Aug  2  2018 tamucchpcmlogin.bash
-rw-rw-rw- 1 cbird cbird  42 Jan 11  2019 oduhpcmlogin.bash
-rw-rw-rw- 1 cbird cbird  61 Feb 15  2019 mntUSB.bash
-rw-rw-rw- 1 cbird cbird  93 Jun 21 06:46 onedrive.bash
drwxrwxrwx 1 cbird cbird 512 Aug 24 10:57 downloads
drwxrwxrwx 1 cbird cbird 512 Aug 24 11:25 CSB
(base) cbird@LAPTOP-URS0LRPO:~$
```

Interpreting Output of `ls -l`

Permissions



User

Group

Global

File

```
-rw-rw-rw-
```

Write

Read

Execute

Dir

```
drwxrwxrwx
```

Paths

- A path is the address of file or directory

- An **absolute path** is complete and starts with root / or a variable that starts with root
  - These return the same result regardless of pwd

  ```
  /home/<username>/CSB
  ~/CSB
  $HOME/CSBB
  ```

- **Relative paths** start from the present location
  - These only work if you are in the right dir
  - . Means present directory
  - .. means parent directory

  ```
  ./CSB
  CSB
  ```

- It's best not to used spaces in dir and file names
  - See pg 21 for dealing w/ spaces

```
$ cd ~

# show contents of CSB dir
# absolute paths
$ ls /home/<username>/CSB
$ ls ~/CSB
$ ls $HOME/CSB

# relative paths
$ ls ./CSB
$ ls CSB
```

Note: if a path includes a space, either wrap path in quotes or precede each space with \

Mind Expander 1.1

Computational Biology

Lecture 1

Dr. Chris Bird

$ Welcome to the Matrix

1.5 Commands to Remember

Computational Biology

Lecture 1

Dr. Chris Bird

Copy with cp <from> <to>

```
# goto sandbox
$ cd ~/CSB/unix/sandbox

# copy the following file to the present directory
$ cp ../data/Buzzard2015_about.txt .

# copy file and rename it in present dir
$ cp ../data/Buzzard2015_about.txt ./Buzzard2015_about2.txt

# copy whole data dir to present dir, then view present dir
$ cp -rf ../data .
$ ls
```

Note: -r means recursive, -f means force

Move or rename with `mv <from> <to>`

```
# make sure you are still in sandbox, if not then cd ~/CSB/unix/sandbox
$ pwd

# move the file to the data directory
$ mv Buzzard2015_about2.txt ../data

# rename a file that isn’t in your pwd
$ mv ../data/Buzzard2015_about2.txt ../data/Buzzard2015_about_new.txt

# check your work
$ ls ../data
```

Note: bash gives no positive feedback, only negative if something is wrong

Create file with `touch <filename>`

```
# make sure you are still in sandbox, if not then cd ~/CSB/unix/sandbox
$ pwd

# inspect the current contents of the directory
$ ls -l

# create a new file (you can list multiple files)
$ touch new_file.txt

# inspect the contents of the directory again
$ ls -l

# if you touch the file a second time, the time of last access will change
$ touch new_file.txt
$ ls -l
```

Note: bash gives no positive feedback, only negative if something is wrong

Remove file(s) or dir(s) with `rm <name>`
Make dir with `mkdir <name>`

```
# make sure you are still in sandbox, if not then cd ~/CSB/unix/sandbox
$ pwd

# delete new_file.txt in sandbox, the –i requests confirmation
$ rm -i new_file.txt

# make dir d1 in present dir, d2 in d1, and d3 in d2; if you have tree try it
$ mkdir -p d1/d2/d3
$ tree d1
d1
└── d2
    └── d3

# remove the d1,d2,& d3 dirs recursively
$ rm -rf d1
```

be careful with `rm`, you could delete your whole computer and there is no undo

View large files with **less -S <filename>**
Print and concatenate files **cat <filename>**
Print and sort files **sort <filename>**

```
# move to the data dir
$ cd ~/CSB/unix/data

# look at DNA alignment file, try duckduckgo search on “bash less commands”
$ less –S Marra2014_data.fasta

# type /ATCG inside of less to search; u=up, d=down, G=end, g=begin, q=exit

# concatenate files and/or print to screen
$ cat Marra2014_about.txt Gesquiere2011_about.txt Buzzard2015_about.txt

# print the sorted lines of a file
$ sort Gesquiere2011_data.csv

# sort numerically by column 2 in reverse order and view in less
$ sort –n –k2 –r Gesquiere2011_data.csv | less
```

Count words with `wc <filename>`

Determine file type `file <filename>`

```
# count lines, words, and characters
$ wc Gesquiere2011_about.txt

# count lines only
$ wc -l Marra2014_about.txt

# determine file type, ASCII is a type of human-readable text file
$ file Marra2014_about.txt
Marra2014_about.txt: ASCII English text
```

Don’t forget to use *Tab* key to autocomplete names, prevents spelling mistakes

Get beginning of file	`head –n # <filename>`
Get end of file	`tail –n # <filename>`

```
# display first two lines of a file
$ head -n 2 Gesquiere2011_data.csv

# display last two lines of file
$ tail -n 2 Gesquiere2011_data.csv

# display from line 2 onward
# (i.e., removing the header of the file)
$ tail -n +2 Gesquiere2011_data.csv

# display all but the last line
$ head -n -1 Gesquiere2011_data.csv
```

Don’t forget to use *Tab* key to autocomplete names, prevents spelling mistakes

Mind Expander 1.2

Computational Biology

Lecture 1

Dr. Chris Bird

$ Welcome to the Matrix

1.6 Advanced Commands

Computational Biology

Lecture 1

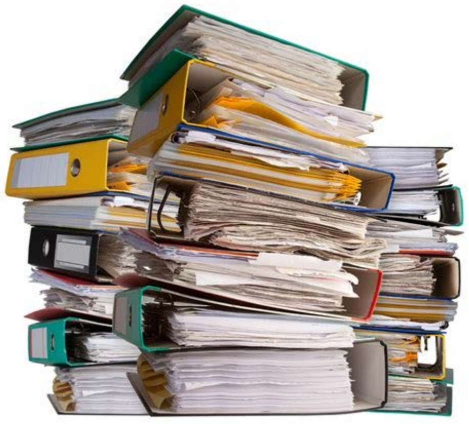
Dr. Chris Bird

Redirection of output (stdout) to file	`[command] > filename`
Append stdout to file	`[command] >> filename`
Redirect contents of file to stdin	`[command] < filename`

```
# let’s start by moving to our sandbox
$ cd ~/CSB/unix/sandbox

# print text to screen, then print to file, then print file to screen
$ echo "My first line"
$ echo "My first line" > test.txt
$ cat test.txt

# append file with additional text, then print file to screen
$ echo "My second line" >> test.txt
$ cat test.txt
```

Don’t forget to use *Tab* key to autocomplete names, prevents spelling mistakes

Problem Solving Scenario



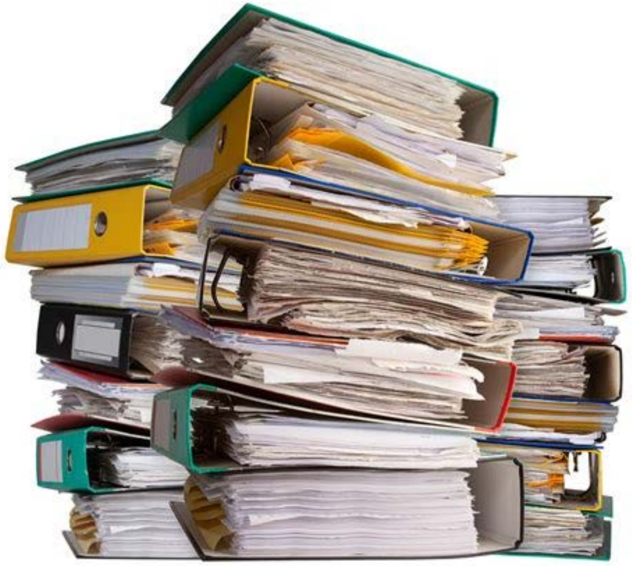
- A machine provides you with thousands of data files
- There’s so many, it’s breaking your file browser
- How many files are there?

- We will use unix/data/Saavedra2013 as an example of a directory with many files

```
# save file names to file in pwd
$ ls ../data/Saavedra2013 > filelist.txt

# look at the file
$ cat filelist.txt

# count lines in a file
$ wc -l filelist.txt

# remove the file
$ rm filelist.txt
```

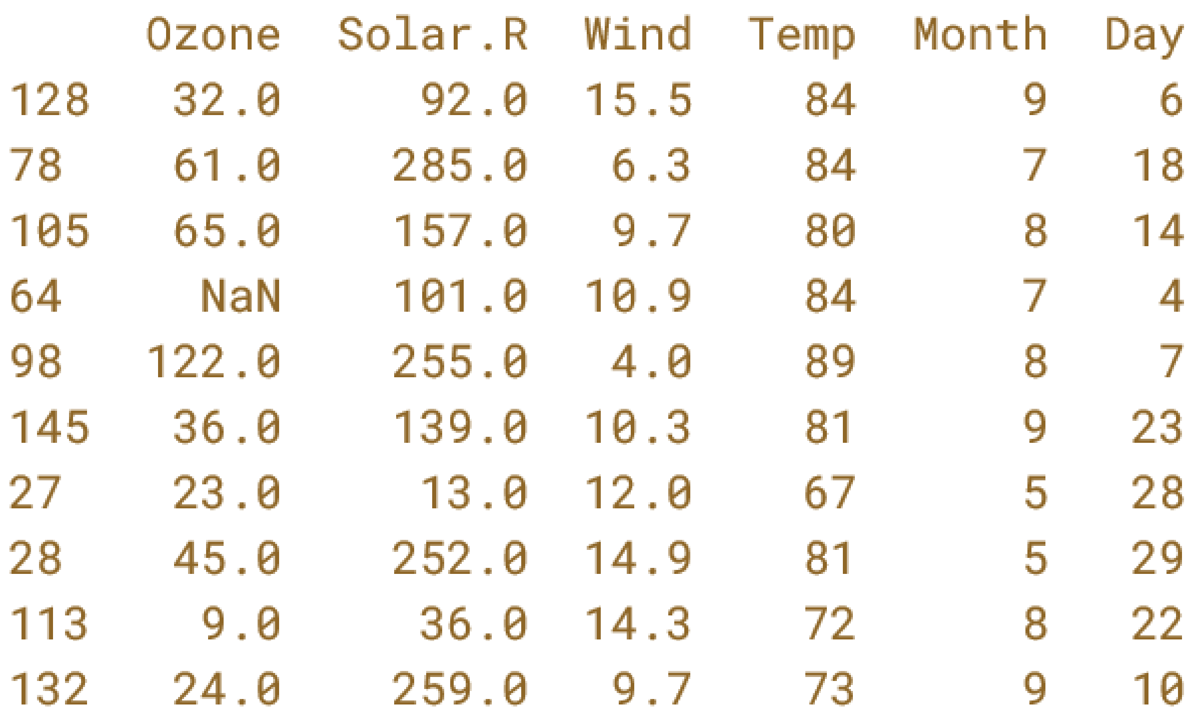
Problem Solving Scenario – Application of pipe |

- A pipe passes the stdout from one command to the stdin of another
- How many files are there?



```
# list file names
$ ls ../data/Saavedra2013

# list file names and pipe into wc
$ ls ../data/Saavedra2013 | wc –l
59
```

TSV and CSV Data Files



```
     Ozone  Solar.R  Wind  Temp  Month  Day
128   32.0     92.0  15.5    84      9    6
78    61.0    285.0   6.3    84      7   18
105   65.0    157.0   9.7    80      8   14
64     NaN    101.0  10.9    84      7    4
98   122.0    255.0   4.0    89      8    7
145   36.0    139.0  10.3    81      9   23
27    23.0     13.0  12.0    67      5   28
28    45.0    252.0  14.9    81      5   29
113    9.0     36.0  14.3    72      8   22
132   24.0    259.0   9.7    73      9   10
```

- Tab Separated Values (TSV)
  - Tabs denote columns
- Comma Separated Values (CSV)
  - Commas denote columns
- Tidy data
  - Each row is one unit of observation
  - Each column is one dimension or aspect of the units of observation
- File extensions not always accurate

It’s Easy to Convert Among Formats Using tr

```
# view contents of csv
$ less -S ../data/Pacifici2013_data.csv

# replace semicolons with commas using tr [find] [replace]
$ cat ../data/Pacifici2013_data.csv | tr “;” “,” | less –S

# view as tsv
# \t is the nearly universal symbol for tab
$ cat ../data/Pacifici2013_data.csv | tr ";" "\t" | less -S
```

tr is short for translate

Using **cut** to grab columns and **head** to grab rows

```
# change directory
$ cd ~/CSB/unix/data

# display first line of file (i.e., header of CSV file)
$ head -n 1 Pacifici2013_data.csv

# display first column of file
$ cut -d ";" –f 1 Pacifici2013_data.csv

# display second through fourth columns
$ cut -d ";" -f 2-4 Pacifici2013_data.csv

# display first “cell” of data
$ head -n 1 Pacifici2013_data.csv | cut -d ";" -f 1
```

Connecting `cut head tail sort uniq`

```
# select 2nd column, display first 5 elements
$ cut -d ";" -f 2 Pacifici2013_data.csv | head -n 5

# select 2nd and 8th columns, display first 3 elements
$ cut -d ";" -f 2,8 Pacifici2013_data.csv | head -n 3

# select 2nd column without header, show 5 first elements
$ cut -d ";" -f 2 Pacifici2013_data.csv | tail -n +2 | head -n 5

# identify the orders in csv
# select 2nd column without header, unique sorted elements
$ cut -d ";" -f 2 Pacifici2013_data.csv | tail -n +2 | sort |\
> uniq

# count how many records per order in csv
$ cut -d ";" -f 2 Pacifici2013_data.csv | tail -n +2 | sort |\
> uniq -c
```

$ Welcome to the Matrix

Questions?

Computational Biology

Lecture 1

Dr. Chris Bird

Mind Expander 1.3

Computational Biology

Lecture 1

Dr. Chris Bird