Introduction to Unix command line & Shell programming

Sergey Knyazev

UCLA Pathology & Laboratory Medicine

knyazev@ucla.edu

2022 Fall

Thanks Dr. Lefaudeux and Dr. Guo for slides

Agenda

- Preparation

- Day 1: Introduction to Unix Command line
  - Navigating through file system
  - Manipulating files and directories

- Day 2: Advanced Command line
  - File processing using awk, grep, sed …

- Day 3: Shell scripting and running jobs on hoffman2
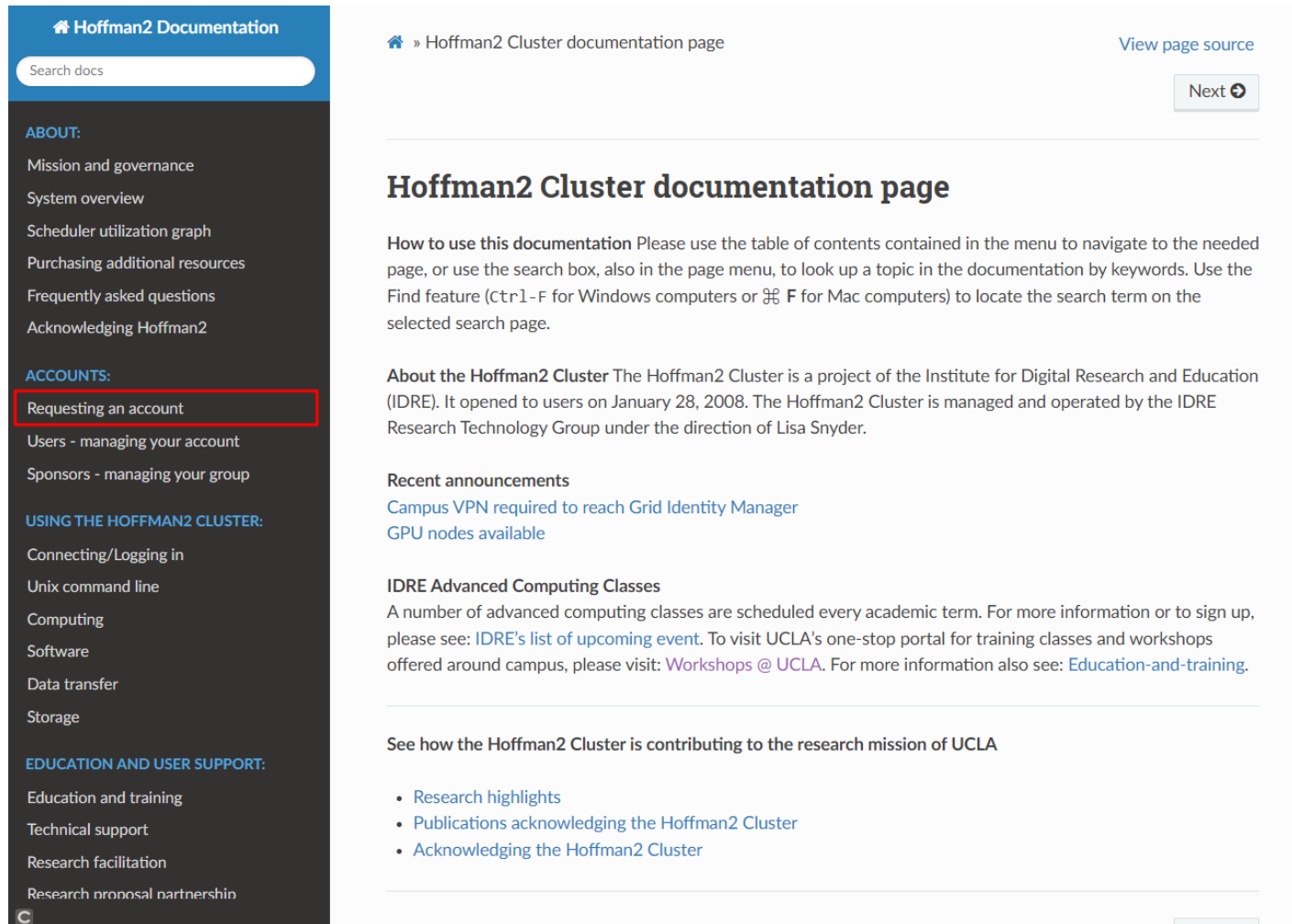
Notations of the slides

- Code chunk starts with "➢", e.g.
  - ➢ echo 'Hello world!'

- Link is underlined

- Practice comes with

Preparation

Step 1: Register a hoffman2 account

- Go to Hoffman2 documentation: https://www.hoffman2.idre.ucla.edu/

Step 1: Register a hoffman2 account

• Click Register here!

(Campus VPN might be required to access the registration website)

Step 1: Register a hoffman2 account

- Fill in the information

(Note: if you don't have a sponsor or your PI is not in the sponsor faculty list, you can choose Collaboratory Workshops)

Step 2: Get your computer ready!

- If you are a Windows user
  - Option 1: Install MobaXterm
    - Download the free software https://mobaxterm.mobatek.net/download.html
    - Install and start local terminal

Step 2: Get your computer ready!

- If you are a Windows user
  - Option 1: Install MobaXterm
    - Download the free software https://mobaxterm.mobatek.net/download.html
    - Install and start local terminal

  - Option 2: Install Windows Subsystem Linux (WSL)
    - Follow this official instructions
    - Open the terminal

Step 2: Get your computer ready!
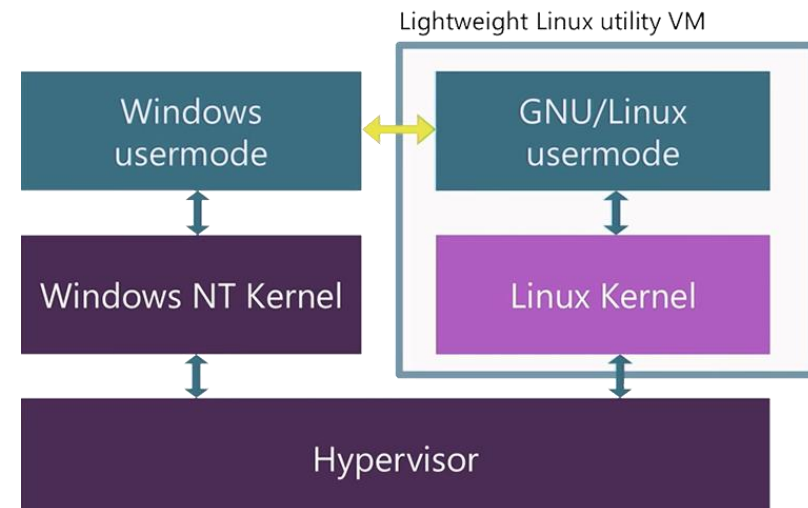
- If you are a Windows user
  - Option 1: Install MobaXterm
    - Download the free software https://mobaxterm.mobatek.net/download.html
    - Install and start local terminal

  - Option 2: Install Windows Subsystem Linux (WSL)
    - Follow this official instructions
    - Open the terminal

  - Option N: Other command line tools like Cygwin, GitBash, Xshell, etc.

Step 2: Get your computer ready!

- If you are a macOS or Linux user
  - Just open the Terminal
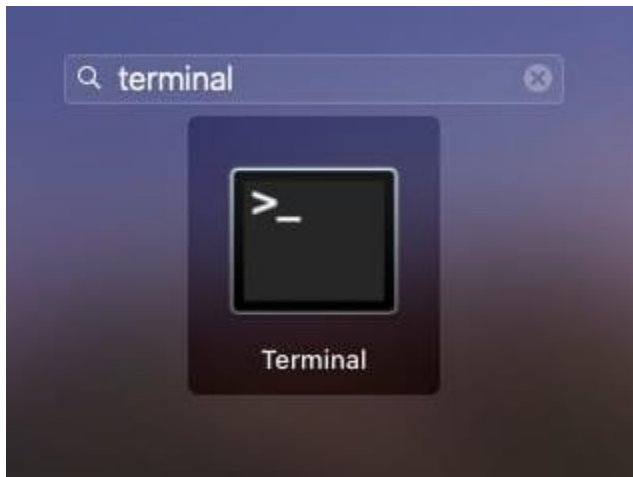
macOS

ubuntu

Questions and references

- Write down questions at this [Google doc](#)

- Reference book is freely available [online](#)
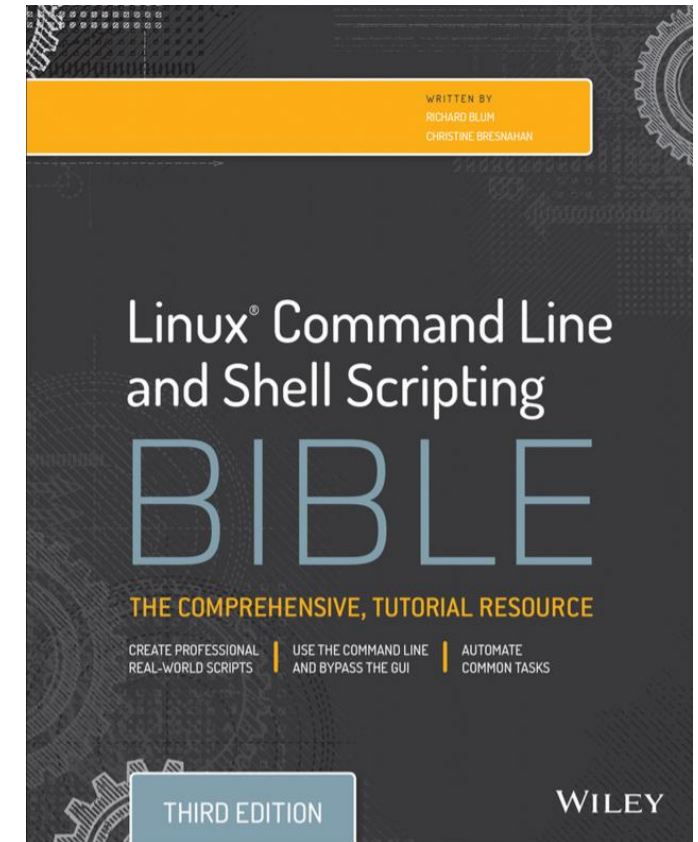
Day 1: Introduction to Unix command line

Overview

Time

- 3-hour workshop (45min + 45min + 30min + practice/Q&A)

Topics

❏ Unix/Linux and Command line interface
  - What's that?
  - Why do we learn?

❏ Log into Hoffman2 cluster

❏ File system and file path

❏ Basic Command lines
  - Navigating through file system
  - Manipulating files and directories
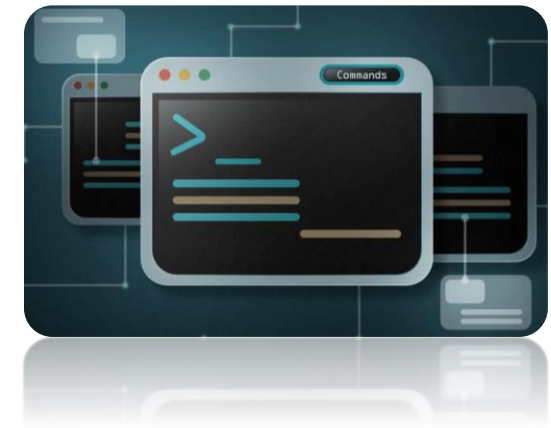
What is Unix/Linux?

- Unix is a family of multitasking, multiuser computer operating systems, developed in the 1970s at the Bell Labs

- Linux is a open-source, Unix-like system

Advantages:
- Secure
- Stable
- Efficient
- Flexible
- Programmer-friendly

…
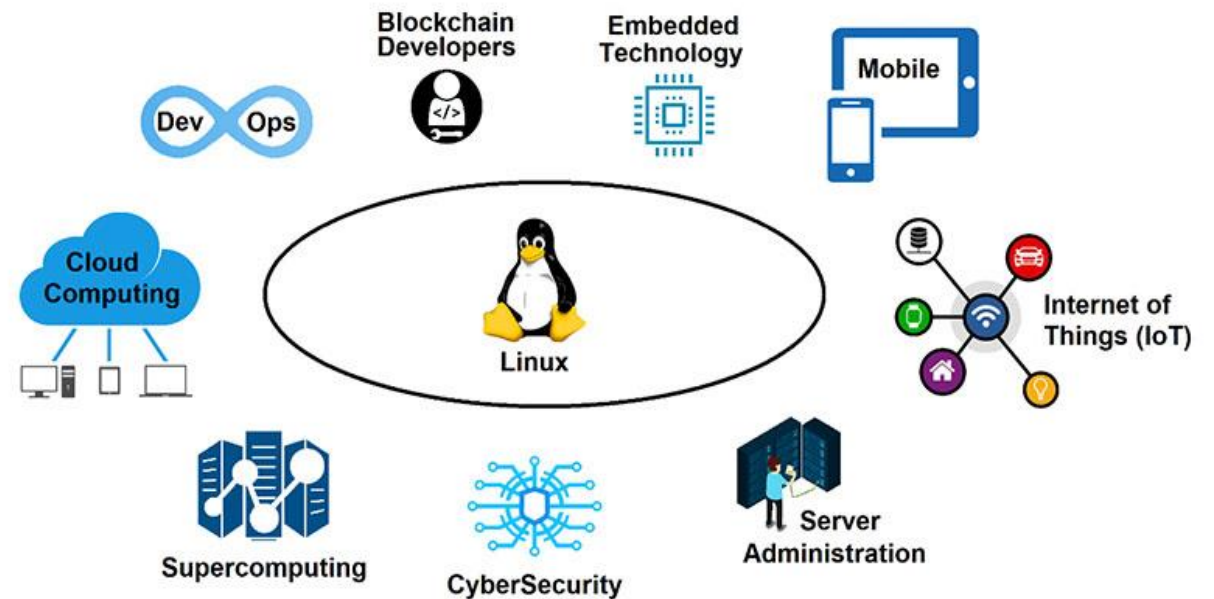
Unix/Linux is widely used in academia and industry!

Why Unix/Linux?

- Unix/Linux rules servers and supercomputers
  - Hoffman2 runs on CentOS

Operating system Family System Share

● Linux

100%
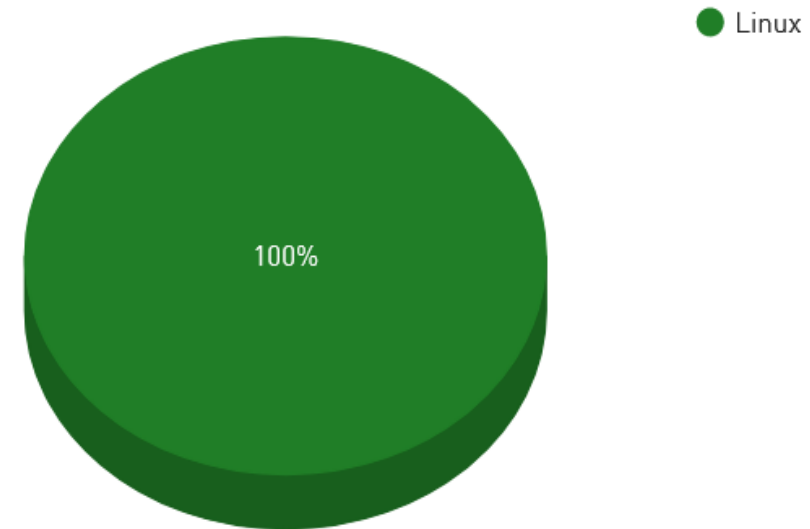
Operating System System Share
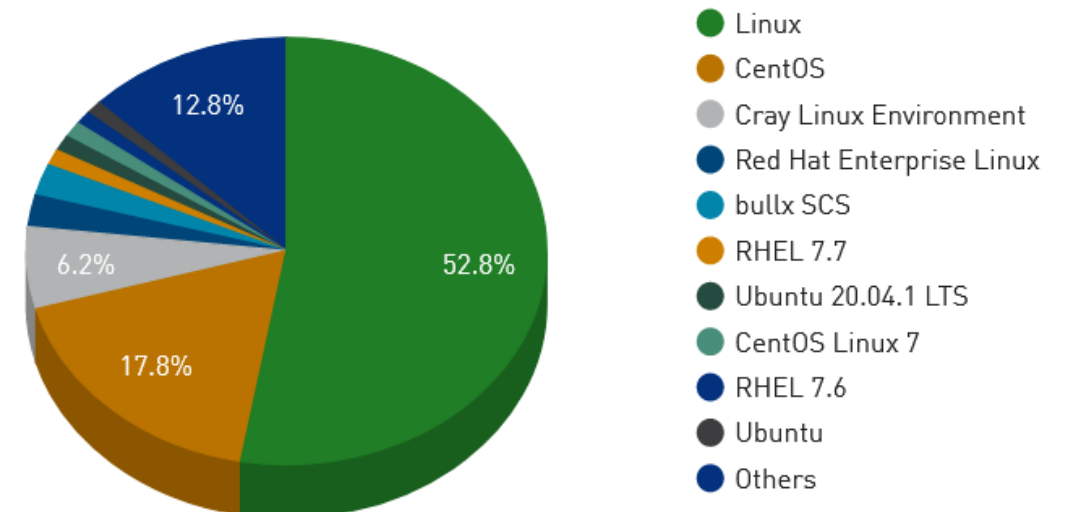
52.8%

17.8%

6.2%

12.8%

● Linux
● CentOS
● Cray Linux Environment
● Red Hat Enterprise Linux
● bullx SCS
● RHEL 7.7
● Ubuntu 20.04.1 LTS
● CentOS Linux 7
● RHEL 7.6
● Ubuntu
● Others

Why Unix/Linux?

- Unix/Linux rules servers and supercomputers

- Data accumulate rapidly in Bioscience
  - Bioinformatics tools are usually developed in Unix/Linux environment
  - We process data in Unix/Linux environment

Moore's law: the number of transistors in microchip doubles every 18 months

NIH; Stephens ZD et al. (2015) *Plos Biology*
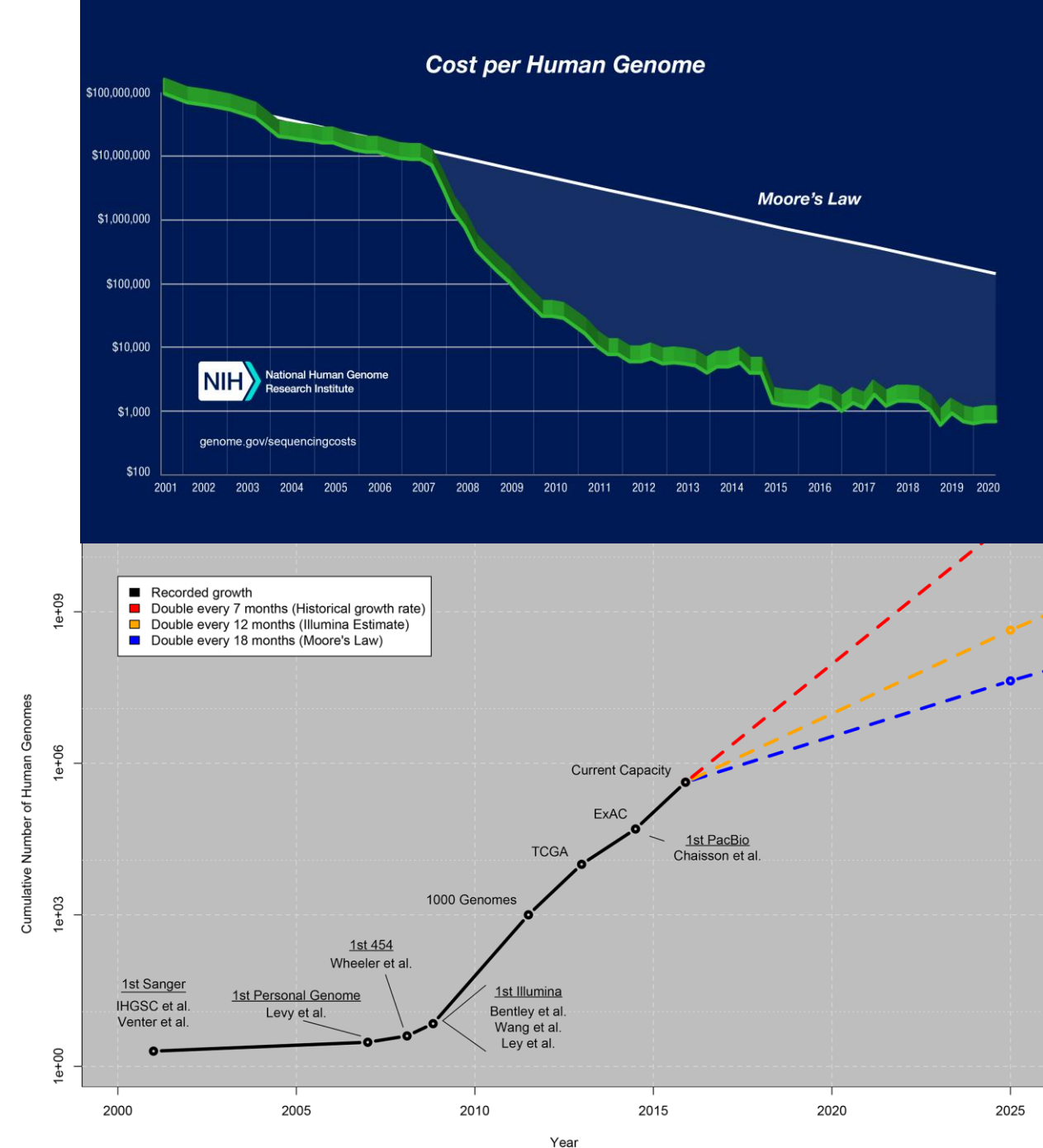
Why Unix/Linux?

- Unix/Linux rules servers and supercomputers

- Data accumulate rapidly in Bioscience

- Unix/Linux is good at repetitive and tedious work
  - It can free your hand and mind
  - It's good for reproducible science
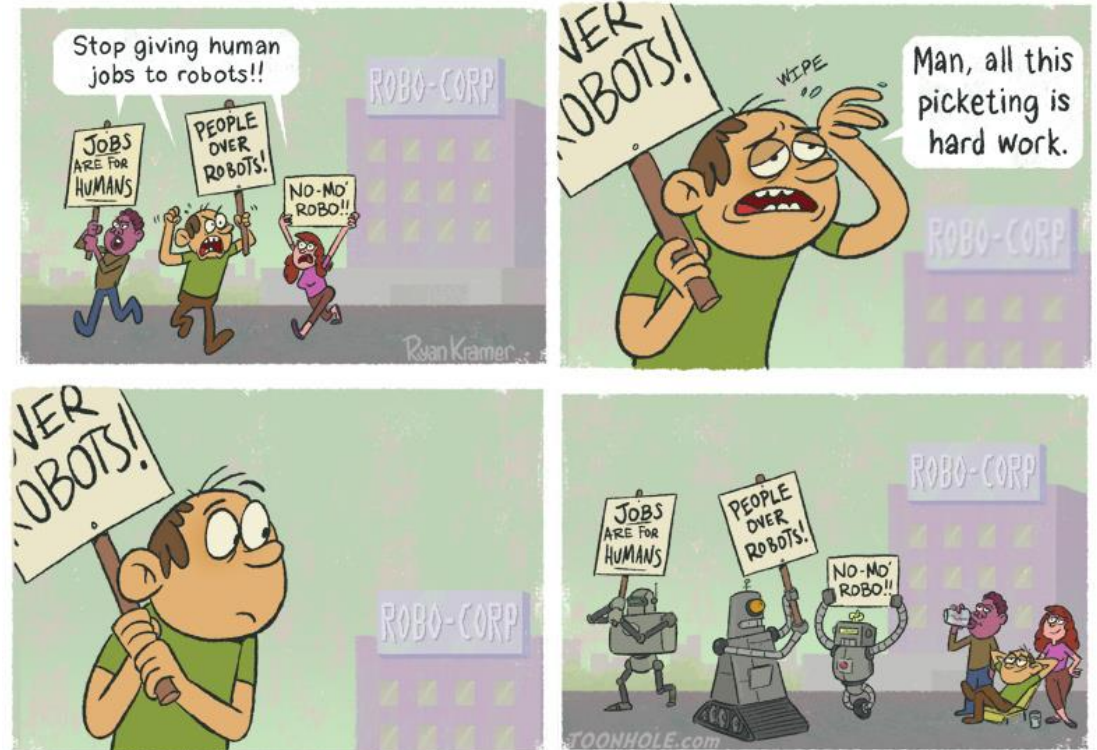
How to give orders to the machine?

Ideal human-computer interaction
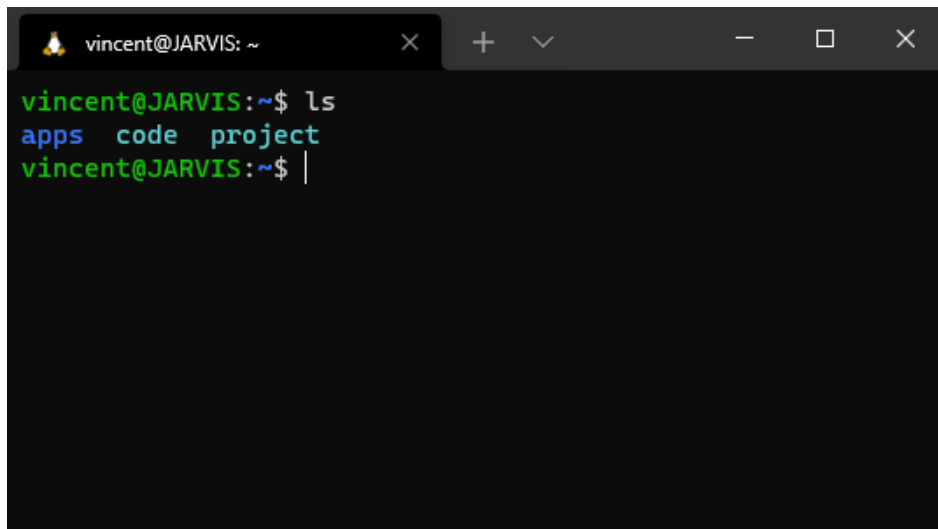
Actual human-computer interaction
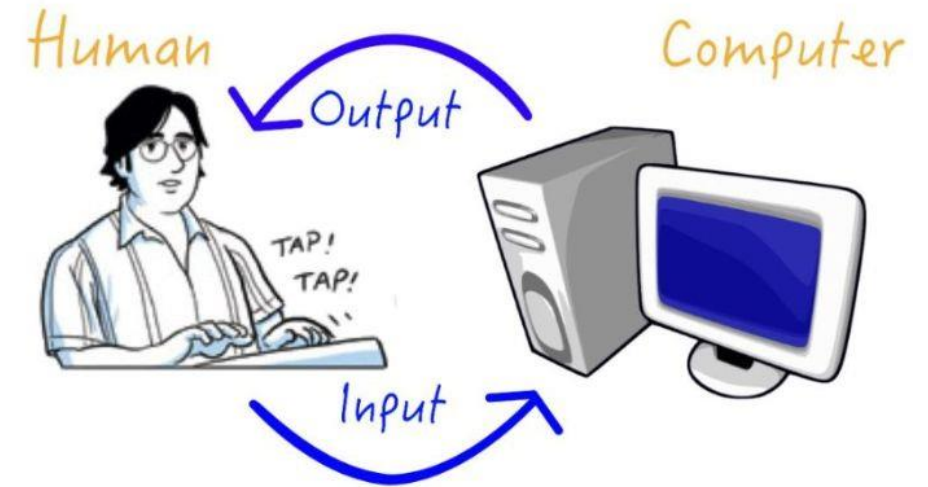
Use Command Line!

Screencap from Ironman movie;
https://www.hiof.no/iio/itk/forskning/grupper/interaction-design/aktuelt/what-is-a-human-computer-interface-.html

Command Line Interface vs Graphical User Interface

Command Line Interface vs Graphical User Interface

	CLI: Command Line Interface	GUI: Graphical User Interface
Ease	Higher degree of memorization and familiarity required	More visually intuitive
Control	More control over files	Advance tasks still may need to require command-line
Resources	Requires less system resources	Requires more system resources
Speed / Strain	Keyboard is utilized	Keyboard + mouse utilized
Scripting	Can create scripts for a sequence of commands	Much less flexible for writing programs
Diversity	Original commands remain the same	Interface can change with new versions

Hoffman2 cluster overview

- Over 1,300 nodes (64-bit)

- Over 20,000 cores

- Over 50 TB aggregated memory

- 2.5 PB storage and 2PB backup

Hoffman2 is currently the largest and most powerful shared cluster in the University of California system.

UCLA user's Terminal

Job submission   Job completion

Login Nodes

Computing Nodes, Storage

Remote server
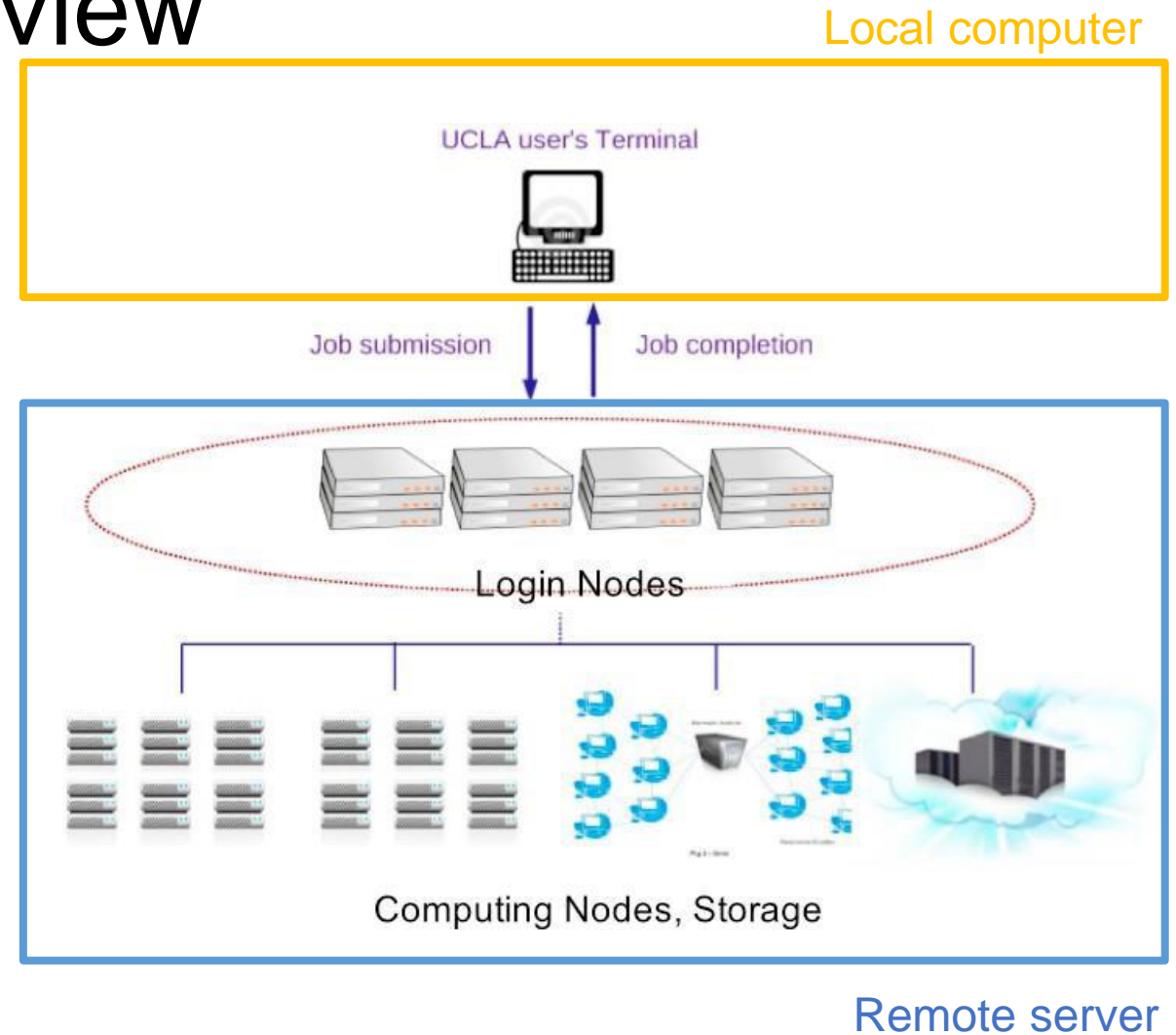
Let's login to hoffman2 from command line interface (CLI)!

22

Access hoffman2 (Windows user)

Start local terminal:
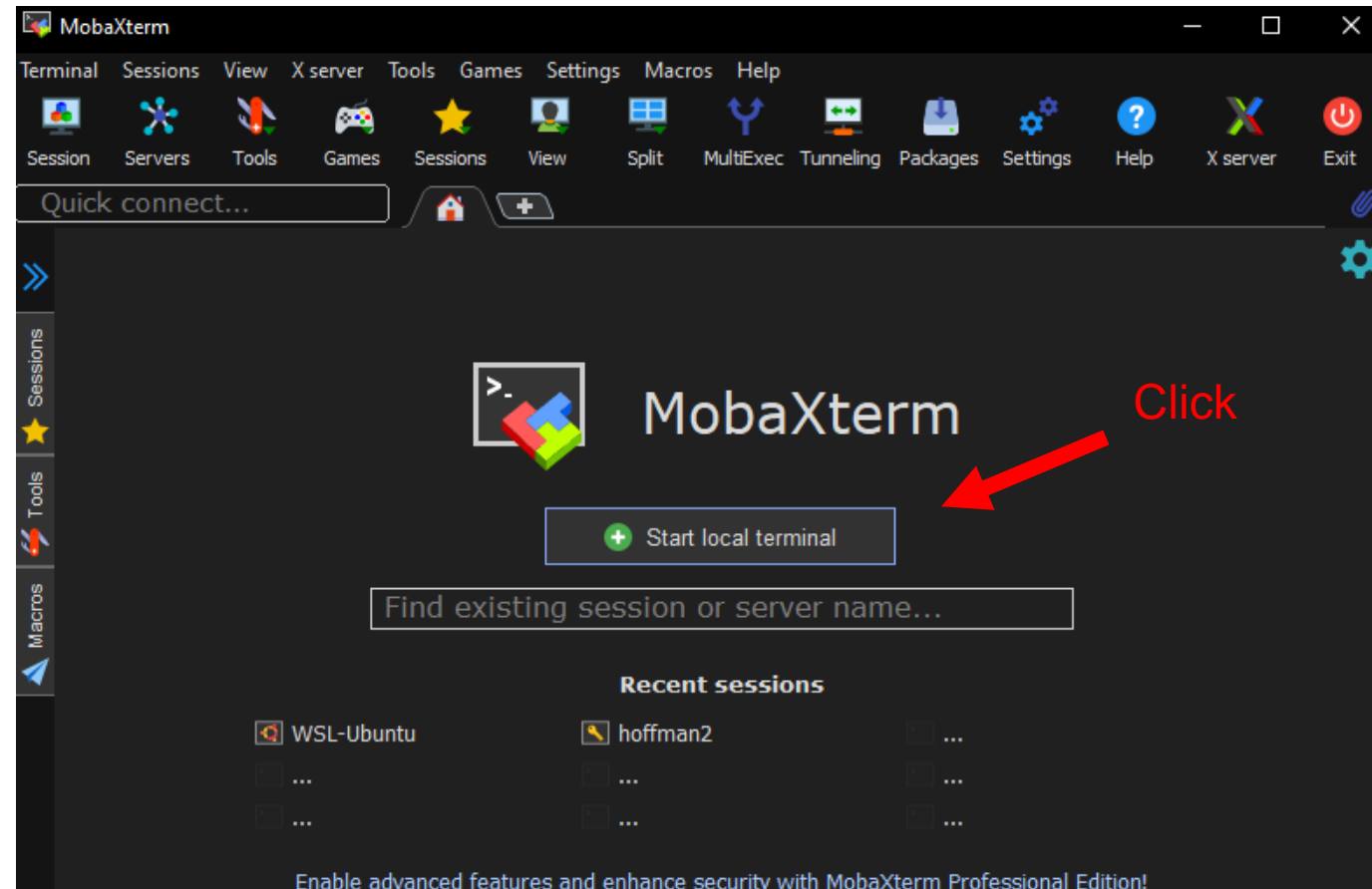
Access hoffman2 (Windows user)

Type in the following commands to your computer terminal:

➢ ssh username@hoffman2.idre.ucla.edu

   (*replace* username *with your hoffman2 username, e.g. wbguo*)
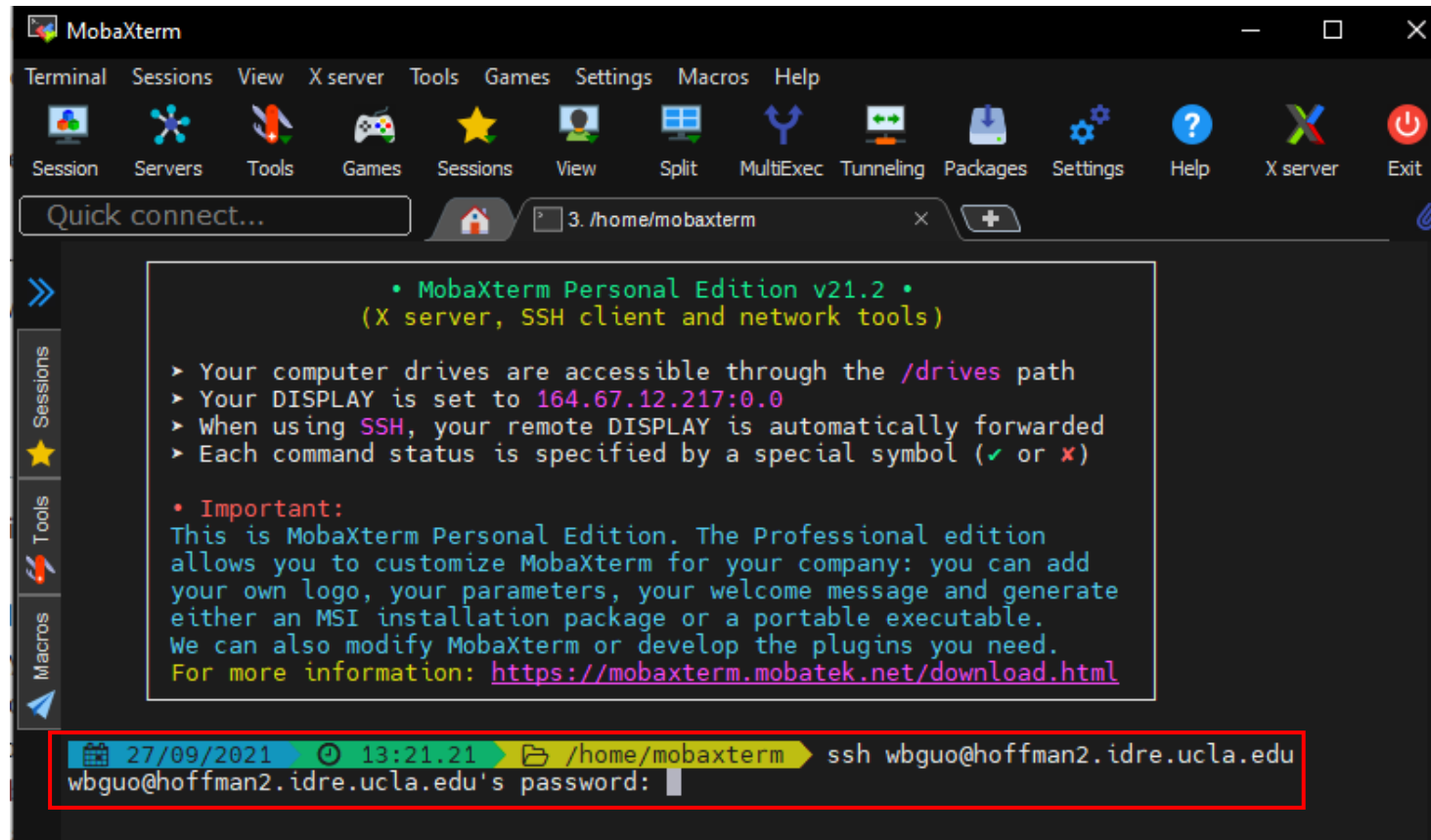
Note:
- Command line is case-sensitive
- For security reason, when you type in password, nothing shows up on the screen

24

Access hoffman2 (Windows user)

Type in the following commands to your computer terminal:

➤ ssh username@hoffman2.idre.ucla.edu

(*replace* username *with your hoffman2 username, e.g. wbguo*)

Access hoffman2 (Windows user)

Another option

- Click Session

Access hoffman2 (Windows user)

Another option
- Click Session
- Click SSH

Access hoffman2 (Windows user)

Another option

- Click Session
- Click SSH
- Fill in the blank
- Click OK

Access hoffman2 (Windows user)

Another option

- Click Session
- Click SSH
- Fill in the blank
- Click OK

Access hoffman2 (macOS or Linux user)

Type in the following commands to your computer terminal:

➢ ssh username@hoffman2.idre.ucla.edu

    (*replace* username *with your hoffman2 username, e.g. wbguo*)

Access hoffman2 (macOS or Linux user)

Type in the following commands to your computer terminal:

➤ ssh username@hoffman2.idre.ucla.edu

(*replace* username *with your hoffman2 username, e.g. wbguo*)

```
●●●          vincent — wbguo@login1:~ — ssh wbguo@hoffman2.idre.ucla.edu — 89×25

Please do NOT compute on the login nodes.

Processes running on the login nodes which seriously degrade others'
use of the system may be terminated without warning.  Use qrsh to obtain
an interactive shell on a compute node for CPU or I/O intensive tasks.

The following news items are currently posted:

    Hoffman2 Cluster OS upgrade status UPDATE
    Jupyter notebooks download a fresh copy of h2jupynb
    Hoffman2 Cluster OS upgrade status UPDATE.bak
    Q Chem version 5.4.1 now available
    How to change the time request on pending jobs
    Nodes available on the next version of the OS
    GPU nodes available
    IDRE Workshops and Training Sessions

Enter shownews to read the full text of a news item.
Attention (2021-09-27): If you have ~/.julia_rh7, you may want to move it to ~/.julia
(For more details, see https://docs.julialang.org/en/v1/manual/environment-variables/#JUL
IA_DEPOT_PATH)
Loading R/4.1.0-BIO
  Loading requirement: curl/7.70.0 R/4.1.0 R/4.1.0-DS
[wbguo@login1 ~]$ ▌
```

Unix Shell

`[wbguo@login1 ~]$` ▮ → Command to type-in

User name   Server name   Folder name

- A **Shell** provides you with an interface to Unix/Linux system

- It waits for your command and will execute it after you press **enter** key

- After finishing, it will return output/error message to the terminal

- If no error message is displayed, it usually means the command was finished successfully (No message is good message)

Practice: login to Hoffman2 and say Hello!

- After login, try
  - ➢ `echo 'Hello world!'`

```
● ● ●          📁 vincent — wbguo@login2:~
[wbguo@login2 ~]$ echo 'Hello world!'
Hello world!
[wbguo@login2 ~]$ ▊
```

  - ➢ `git clone` https://github.com/sergey-knyazev/qcbio-intro2Unix.git

File system

- Appears as hierarchical tree structure
- Consists of files, directories, and symbolic links
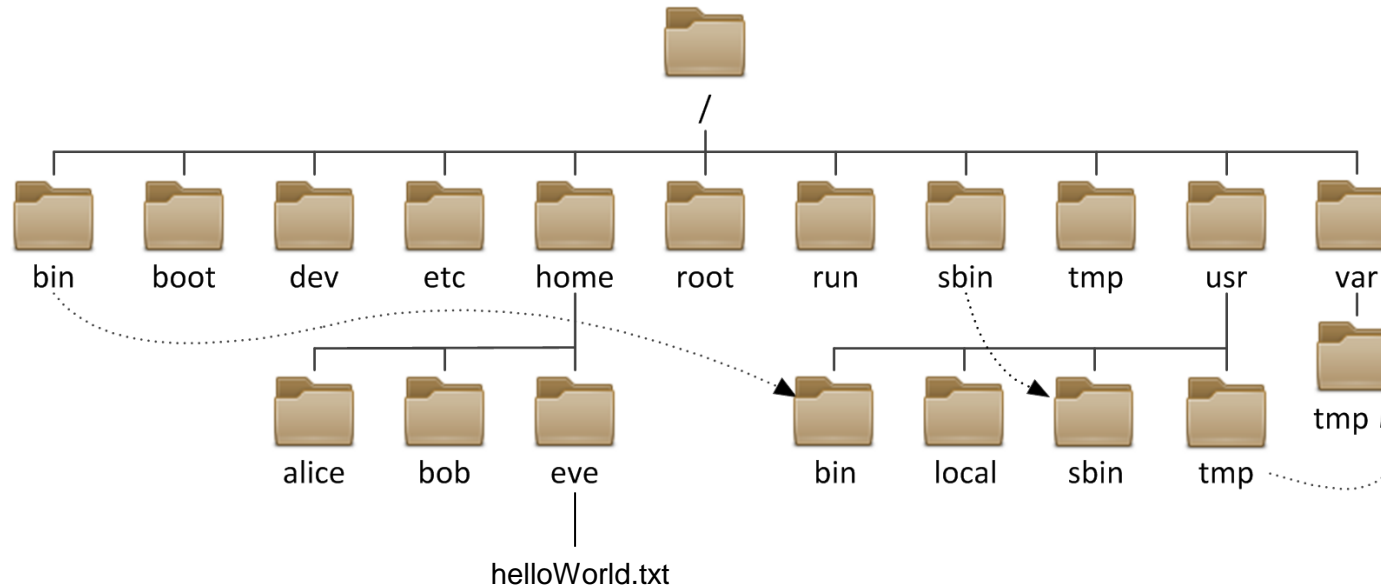
Parent folder: the folder that is **one level up from the current directory**

Path to a file/directory

You can refer to a file/directory using absolute path or relative path

- Absolute path: the full path from the root to the file/directory

  e.g. absolute path: `/home/eve/helloWorld.txt`

- Relative path: relative to your current location

  e.g. relative path: `eve/helloWorld.txt`
  (assume you are in folder `/home`)
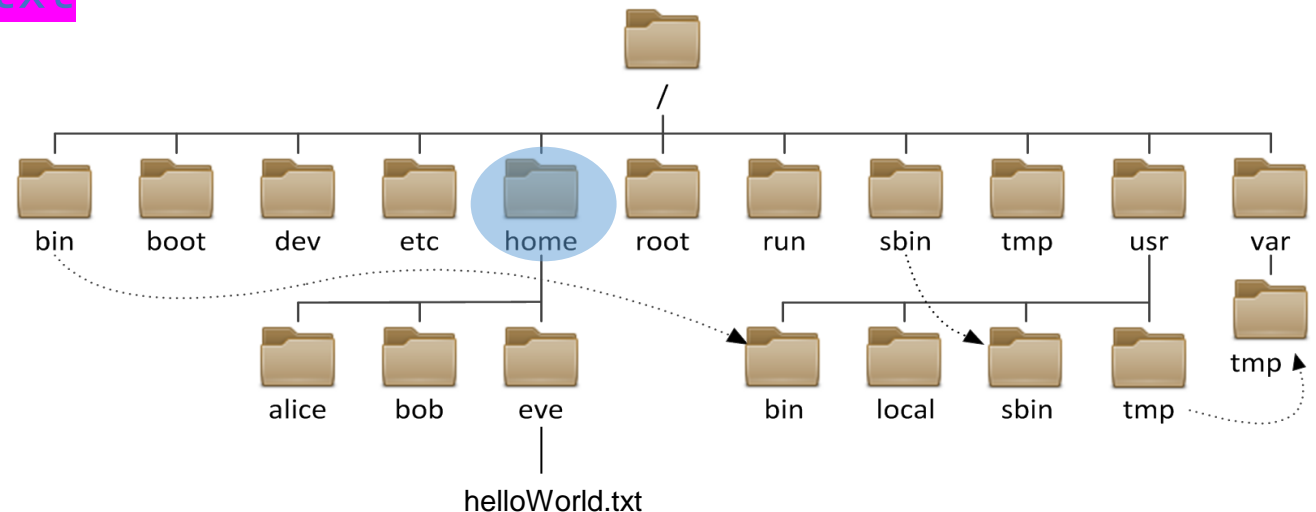
Navigate through the file system

- pwd
- cd
- ls
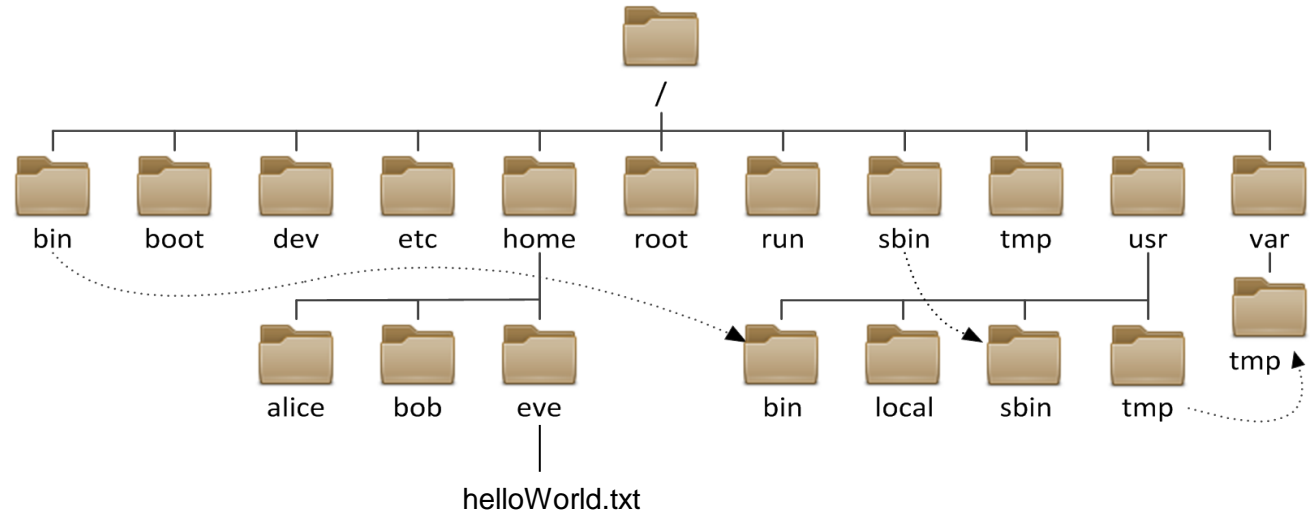
Command pwd

- pwd: **p**rint **w**orking **d**irectory (show the absolute path of current directory)

- Syntax: pwd

```
[wbguo@login1 ~]$ pwd
/u/home/w/wbguo
```

Note: When you open a shell, you start in your home directory ~

Command `cd`

- `cd`: **c**hange **d**irectory (go to another directory)

- Syntax: `cd <directory path>`

```
[wbguo@login1 ~]$ cd qcbio-intro2Unix/
[wbguo@login1 qcbio-intro2Unix]$ pwd
/u/home/w/wbguo/qcbio-intro2Unix
```

If you don't have a folder called qcbio-intro2Unix, run
➢ `git clone https://github.com/wbvguo/qcbio-intro2Unix.git`

Command `cd`

- **~** is the home directory
  ```
  [wbguo@login1 ~]$ pwd
  /u/home/w/wbguo
  ```

- **.** is the current directory
  ```
  [wbguo@login1 ~]$ cd qcbio-intro2Unix/
  [wbguo@login1 qcbio-intro2Unix]$ cd .
  [wbguo@login1 qcbio-intro2Unix]$
  ```

- **..** is the parent folder of the current directory
  ```
  [wbguo@login1 qcbio-intro2Unix]$ pwd
  /u/home/w/wbguo/qcbio-intro2Unix
  [wbguo@login1 qcbio-intro2Unix]$ cd ..
  [wbguo@login1 ~]$ pwd
  /u/home/w/wbguo
  ```

Practice

- Print absolute path of your home directory

- Go to the qcbio-intro2Unix folder

Command `ls`

- ls: list (list the information of files/folders in a directory)

- Syntax: `ls -<flag> <file_name/directory>`

```
[wbguo@login1 ~]$ cd qcbio-intro2Unix/
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md  day1  day2  day3
[wbguo@login1 qcbio-intro2Unix]$ ls ~
R       iproject   project-collab  project-msehl    qrsh4h8g   run_kde_sim.sh  test
apps    perl5      project-mcdb    qcbio-intro2Unix  qsubJN     sample_list     ttest
```

Command `ls`

flag options:

- `-l`: long list
  ```
  [wbguo@login1 qcbio-intro2Unix]$ ls -l
  total 12
  -rw-r--r--. 1 wbguo matteop   19 Jan  3 22:19 README.md
  drwxr-xr-x. 2 wbguo matteop 4096 Jan  3 22:19 day1
  drwxr-xr-x. 2 wbguo matteop 4096 Jan  3 22:19 day2
  drwxr-xr-x. 3 wbguo matteop 4096 Jan  3 22:19 day3
  ```

- 1st column: permission
- 2nd column: the number of hard links to the file
- 3rd column: owner
- 4th column: group
- 5th column: size
- 6th -8th column: modified time
- 9th column: name

Command `ls`

flag options:

- `-a`: all files/folders

```
[wbguo@login1 qcbio-intro2Unix]$ ls -la
total 28
drwxr-xr-x.  6 wbguo matteop 4096 Jan  3 22:19 .
drwx------. 17 wbguo matteop 8192 Jan  3 22:31 ..
drwxr-xr-x.  8 wbguo matteop 4096 Jan  3 22:19 .git
-rw-r--r--.  1 wbguo matteop   19 Jan  3 22:19 README.md
drwxr-xr-x.  2 wbguo matteop 4096 Jan  3 22:19 day1
drwxr-xr-x.  2 wbguo matteop 4096 Jan  3 22:19 day2
drwxr-xr-x.  3 wbguo matteop 4096 Jan  3 22:19 day3
```

Command `ls`

flag options:

- **-S**: sort by size (big to small)

  ```
  [wbguo@login1 qcbio-intro2Unix]$ ls -lS
  total 12
  drwxr-xr-x. 2 wbguo matteop 4096 Jan  3 22:19 day1
  drwxr-xr-x. 2 wbguo matteop 4096 Jan  3 22:19 day2
  drwxr-xr-x. 3 wbguo matteop 4096 Jan  3 22:19 day3
  -rw-r--r--. 1 wbguo matteop   19 Jan  3 22:19 README.md
  ```

- **-t**: sort by time (from new to old)

  ```
  [wbguo@login1 qcbio-intro2Unix]$ ls -lt
  total 12
  drwxr-xr-x. 3 wbguo matteop 4096 Jan  3 22:19 day3
  drwxr-xr-x. 2 wbguo matteop 4096 Jan  3 22:19 day2
  drwxr-xr-x. 2 wbguo matteop 4096 Jan  3 22:19 day1
  -rw-r--r--. 1 wbguo matteop   19 Jan  3 22:19 README.md
  ```

Command `ls`

flag options:

- **-h**: human readable

```
[wbguo@login1 qcbio-intro2Unix]$ ls -lSh
total 12K
drwxr-xr-x. 2 wbguo matteop 4.0K Jan  3 22:19 day1
drwxr-xr-x. 2 wbguo matteop 4.0K Jan  3 22:19 day2
drwxr-xr-x. 3 wbguo matteop 4.0K Jan  3 22:19 day3
-rw-r--r--. 1 wbguo matteop   19 Jan  3 22:19 README.md
```

- **-r**: reverse order

```
[wbguo@login1 qcbio-intro2Unix]$ ls -lShr
total 12K
-rw-r--r--. 1 wbguo matteop   19 Jan  3 22:19 README.md
drwxr-xr-x. 3 wbguo matteop 4.0K Jan  3 22:19 day3
drwxr-xr-x. 2 wbguo matteop 4.0K Jan  3 22:19 day2
drwxr-xr-x. 2 wbguo matteop 4.0K Jan  3 22:19 day1
```

Tips

- Go back to previous command, use up and down arrows

- Use Tab key for auto completion

- Use `history` to see your command history

- Use `clear` to clean screen
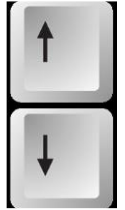
Practice

- Print absolute path of your home directory

- Go to the qcbio-intro2Unix folder

- List the files/folder in long form, sorted by size in human-readable mode

- Go back to home directory

- Clean terminal screen

- Try up and down arrow key and Tab auto-completion

File permission

```
[wbguo@login1 qcbio-intro2Unix]$ ls -l
total 12
-rw-r--r--. 1 wbguo matteop   19 Jan  3 22:19 README.md
drwxr-xr-x. 2 wbguo matteop 4096 Jan  3 22:19 day1
```

File type

Group

Owner   Others
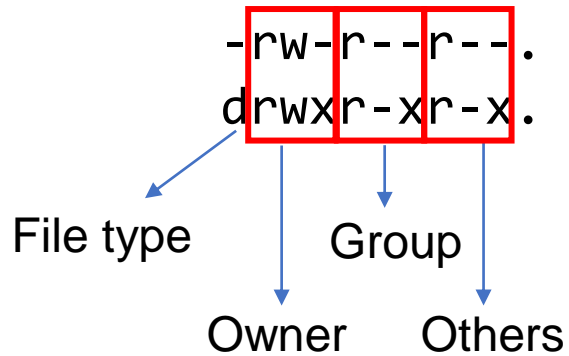
Owners assigned Permission On Every File and Directory

Read
Write
Execute

User

Read
Write
Execute

Group

Read
Write
Execute

All
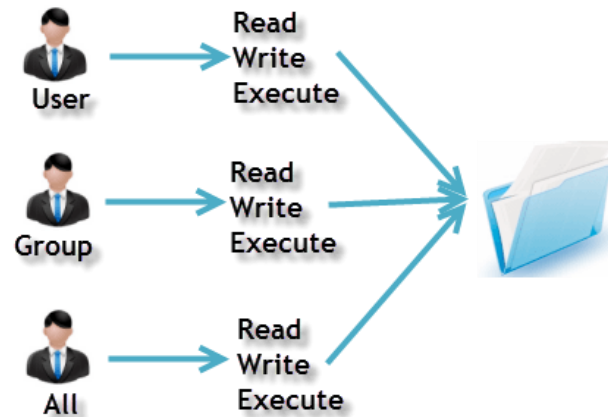
Unix/Linux classified users into 3 types, each can have different permission on the file, which creates a secure environment

File permission

```
[wbguo@login1 qcbio-intro2Unix]$ ls -l
total 12
-rw-r--r--. 1 wbguo matteop   19 Jan  3 22:19 README.md
drwxr-xr-x. 2 wbguo matteop 4096 Jan  3 22:19 day1
```

File type
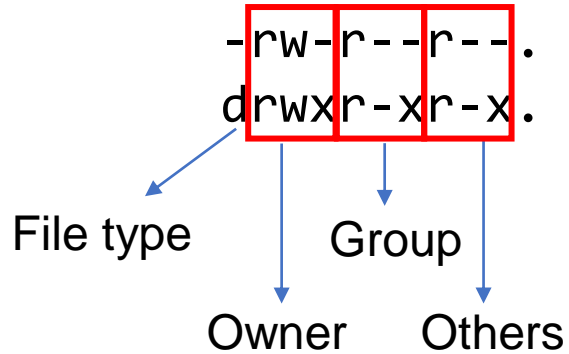
Group

Owner   Others

- File type: d (directory), l (link), - (file)
- User type: Owner (you), Group (users in the same group), Others (the rest of people on the server system)
- Permission type: r (read, encode as 4), w (write, encode as 2), x (execute, encode as 1)
- `rwxr--r--` is equal to 744, meaning Owner/Group/Others can read, but only Owner can write and execute

Change file permission

- chmod: change mode (modify file permission)
- Syntax
  - chmod [user/group/others/all] [+-] [permission] <file/directory name>

Add permission   Remove permission

Assign executive permission to owner
➢ chmod 744 README.md
➢ chmod u+x README.md

```
[wbguo@login1 qcbio-intro2Unix]$ ls -lh
total 12K
-rw-r--r--. 1 wbguo matteop   19 Jan  3 22:19 README.md
drwxr-xr-x. 2 wbguo matteop 4.0K Jan  3 22:19 day1
drwxr-xr-x. 2 wbguo matteop 4.0K Jan  3 22:19 day2
drwxr-xr-x. 3 wbguo matteop 4.0K Jan  3 22:19 day3
[wbguo@login1 qcbio-intro2Unix]$ chmod u+x README.md
[wbguo@login1 qcbio-intro2Unix]$ ls -lh
total 12K
-rwxr--r--. 1 wbguo matteop   19 Jan  3 22:19 README.md
drwxr-xr-x. 2 wbguo matteop 4.0K Jan  3 22:19 day1
drwxr-xr-x. 2 wbguo matteop 4.0K Jan  3 22:19 day2
drwxr-xr-x. 3 wbguo matteop 4.0K Jan  3 22:19 day3
[wbguo@login1 qcbio-intro2Unix]$
```

Practice

- Give README.md execution permission

- Convert day1 to a private folder (group/others users cannot read/write/execute)

Manipulating files and directories

Operations

- Create/Delete
- Copy/Move
- View/Modify
- Compress/Decompress
- Download/Upload
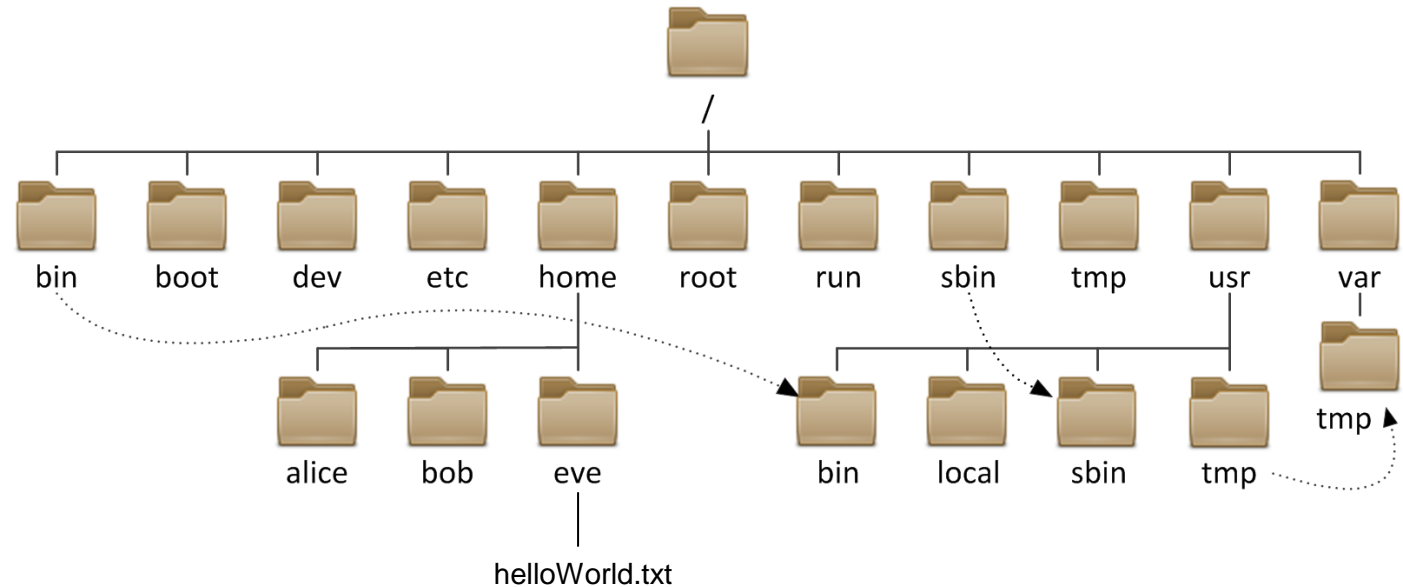
General syntax:
command -<flags> <file/directory>

e.g. ls -lah helloWorld.txt

Create (`mkdir`, `touch`)

- Directory
  - `mkdir`: make directory
  - Syntax: `mkdir <directory path>`
- File
  - `touch`
  - Syntax: `touch <file_name>`

```
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md    day1    day2    day3
[wbguo@login1 qcbio-intro2Unix]$ mkdir myFolder
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md    day1    day2    day3    myFolder
[wbguo@login1 qcbio-intro2Unix]$ touch myFile.txt
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md    day1    day2    day3    myFile.txt    myFolder
```

Create (`mkdir`, `touch`)

- Directory
  - `mkdir`: make directory
  - Syntax: `mkdir <directory path>`
- File
  - `touch`
  - Syntax: `touch <file_name>`

- When file/directory already exists

```
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md   day1   day2   day3   myFile.txt   myFolder
[wbguo@login1 qcbio-intro2Unix]$ mkdir myFolder
mkdir: cannot create directory 'myFolder': File exists
[wbguo@login1 qcbio-intro2Unix]$ touch myFile.txt
```

touch will change the file's modified time when the file exists

Delete (rm)

- rm: remove
- Syntax: `rm -<flag> <directory path>`

- Flag options:
  - -r: recursively
    Remove directory and their contents
  - -i: interactively
    Ask to confirm before every removal
  - -v: verbose
    Explain what is being done
  - -f: force
    Remove without asking

```
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md  day1  day2  day3  myFile.txt  myFolder
[wbguo@login1 qcbio-intro2Unix]$ rm -r myFolder/
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md  day1  day2  day3  myFile.txt
[wbguo@login1 qcbio-intro2Unix]$ rm myFile.txt
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md  day1  day2  day3
```

```
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md  day1  day2  day3  myFile.txt  myFolder
[wbguo@login1 qcbio-intro2Unix]$ rm -ir myFolder/
rm: remove directory 'myFolder/'? y
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md  day1  day2  day3  myFile.txt
[wbguo@login1 qcbio-intro2Unix]$ rm -i myFile.txt
rm: remove regular empty file 'myFile.txt'? n
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md  day1  day2  day3  myFile.txt
```

Delete (`rm`)

- Be cautious with delete!
  - Usually cannot be undone if there is no back-up (no trash folder!)
  - If you accidentally delete files on hoffman2, check out this [hoffman2 recover option](#) or contact [IDRE hoffman2 support](#)

- Back up your important files
  - Google Drive has unlimited capacity for edu users

- Change file to read-only file with `chmod`
  - `chmod 444 <file name>`

Practice

- Create a folder named `myFolder`, a file named `myFile.txt`

- remove both the folder and file in interactive mode

Copy (cp)

- cp: copy
- Syntax: `cp -<flag> <source> <destination>`

- Flag options:
  - -r: recursively

    Copy directories and their contents
  - -i: interactively

    Ask to confirm before overwrite
  - -v: verbose

    Explain what is being done

Copy (cp)

- cp: copy
- Syntax: `cp -<flag> <source> <destination>`

- Copy a file

```
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md   day1   day2   day3
[wbguo@login1 qcbio-intro2Unix]$ cp README.md readme_copy.md
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md   day1   day2   day3   readme_copy.md
```

- Copy a directory

```
[wbguo@login1 qcbio-intro2Unix]$ cp -r day1/ day1_backup
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md   day1   day1_backup   day2   day3   readme_copy.md
```

- Copy a file into directory

```
[wbguo@login1 qcbio-intro2Unix]$ ls day1_backup/
test
[wbguo@login1 qcbio-intro2Unix]$ cp README.md day1_backup/
[wbguo@login1 qcbio-intro2Unix]$ ls day1_backup/
README.md   test
```

Move (mv)

- mv: move
- Syntax: `mv <source> <destination>`

- Rename a file/folder

```
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md   day1   day1_backup   day2   day3   readme_copy.md
[wbguo@login1 qcbio-intro2Unix]$ mv readme_copy.md tmp.md
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md   day1   day1_backup   day2   day3   tmp.md
[wbguo@login1 qcbio-intro2Unix]$ mv day1_backup/ tmp
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md   day1   day2   day3   tmp   tmp.md
```

- Move a file to folder

```
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md   day1   day2   day3   tmp   tmp.md
[wbguo@login1 qcbio-intro2Unix]$ ls tmp
README.md   test
[wbguo@login1 qcbio-intro2Unix]$ mv tmp.md tmp
[wbguo@login1 qcbio-intro2Unix]$ ls tmp/
README.md   test   tmp.md
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md   day1   day2   day3   tmp
```

Practice

- Copy *day1* to *day1_backup*

- Copy `README.md` to `readme_copy.md`, and move it to *day1_backup*

- Rename `readme_copy.md` as `tmp.md`

View (cat)

- cat: concatenate files and print on standard output
- Syntax: `cat <file name>`

```
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md   day1   day2   day3   tmp
[wbguo@login1 qcbio-intro2Unix]$ ls day1/
test
[wbguo@login1 qcbio-intro2Unix]$ cat day1/test
hello world
```

Modify (vi)

- vi: Visual Editor (default text editor for Unix-like system)
- vim: vi improved (an improved text editor tool)
- Syntax: `vi <file name>`

```
[wbguo@login1 qcbio-intro2Unix]$ cd day1/
[wbguo@login1 day1]$ ls
test
[wbguo@login1 day1]$ vi test █
```

```
hello world
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"test" 1L, 12C
```

Modify (vi)

Once inside the file

- Press "**i**" to type in new text (**i**nsertion mode)
- Press "**Esc**" to jump out the insertion mode
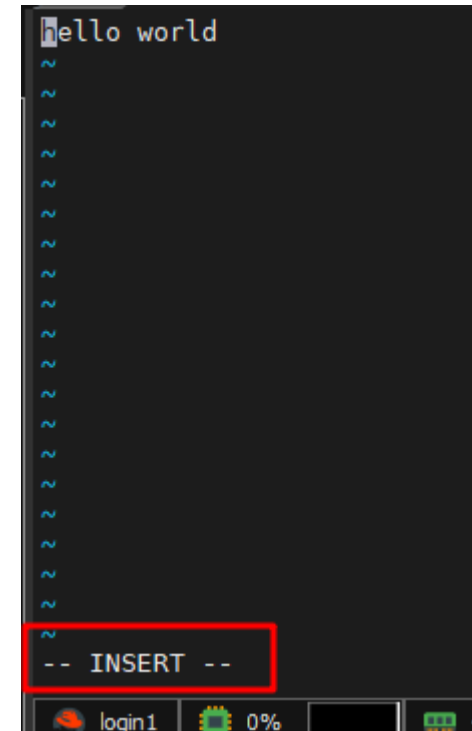
Modify (vi)

Once inside the file

- Press "**i**" to type in new text (**i**nsertion mode)
- Press "**Esc**" to jump out the insertion mode

- Enter
  - :w (write)
  - :q (quit)
  - :wq (write and quit)
  - :q! (force quit – the modification will not be saved)

```
hello world
Command line is cool!
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

Modify (vi)

Once inside the file

- Press "**i**" to type in new text (**i**nsertion mode)
- Press "**Esc**" to jump out the insertion mode

```
[wbguo@login1 day1]$ vi test
[wbguo@login1 day1]$ cat test
hello world
Command line is cool!
```

- Enter
  - :w (write)
  - :q (quit)
  - :wq (write and quit)
  - :q! (force quit – the modification will not be saved)

- Note: if the file exists, then vi editing can modify the file, if the file does not exists, vi editing can create the file

Practice

- Append a line "UCLA QCB wrokshop" to the end of `test` file in *day1* directory

- Use `cat` to display the modified file

- Create a new file using vi, with contents "Less is more"

Compress and decompress (`gzip`)

File

- gzip: GNU zip

- Syntax:
  - Compress: `gzip <file name>`
  - Decompress: `gzip -d <file name>.gz`

- Option:
  - `-d`: decompress

```
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md   day1   day2   day3   tmp
[wbguo@login1 qcbio-intro2Unix]$ chmod u-x README.md
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md   day1   day2   day3   tmp
[wbguo@login1 qcbio-intro2Unix]$ gzip README.md
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md.gz   day1   day2   day3   tmp
[wbguo@login1 qcbio-intro2Unix]$ gzip -d README.md.gz
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md   day1   day2   day3   tmp
```

Compress and decompress (`tar`)

Directory

- `tar`: tape archive (create a compresed archive)
- Syntax:
  - Compress: `tar –czvf <file name>.tar.gz <file name>`
  - Decompress: `tar –xzvf <file name>.tar.gz`

- Options
  - -c: create a new archive
  - -x: extract from archive
  - -v: verbose, list the processed files
  - -f: use archive file
  - -z: compress/extract using gzip

Compress and decompress (`tar`)

Directory

- `tar`: **t**ape **ar**chive (create a compresed archive)

- Syntax:
  - Compress: `tar -czvf <file name>.tar.gz <file name>`
 - Decompress: `tar -xzvf <file name>.tar.gz`

```
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md   day1   day2   day3   tmp
[wbguo@login1 qcbio-intro2Unix]$ tar -czvf tmp.tar.gz tmp/
tmp/
tmp/test
tmp/README.md
tmp/tmp.md
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md   day1   day2   day3   tmp   tmp.tar.gz
```

Download & upload (`scp`)

- scp: openssh secure file copy
- Syntax (on local computer terminal):
  - Upload: `scp <local file> username@hoffman2.idre.ucla.edu:<path>`
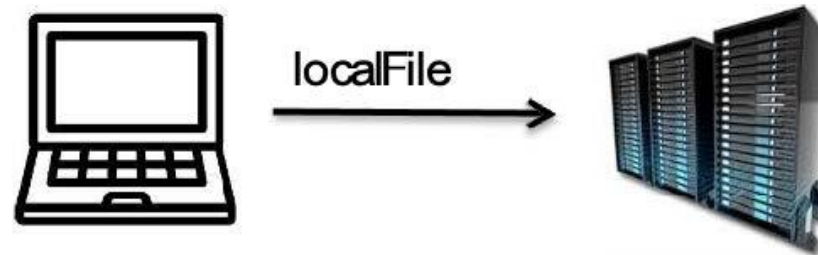
  - Download: `scp username@hoffman2.idre.ucla.edu:<path> .`
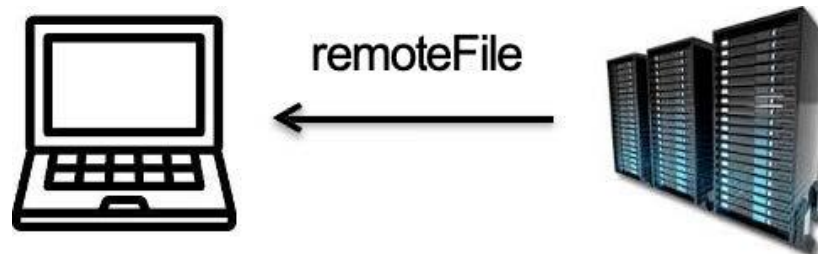
Download & upload (`scp`)

- scp: openssh secure file copy

- Upload

```
[vincent@Wenbins-MacBook-Pro ~ % ls
Applications Downloads    Movies        Pictures      gDrive
Desktop        Dropbox      Music         Public        opt
Documents      Library      OneDrive      Zotero        readme
[vincent@Wenbins-MacBook-Pro ~ % cat readme
this is a local file
[vincent@Wenbins-MacBook-Pro ~ % scp readme wbguo@hoffman2.idre.ucla.edu:qcbio-intro2Unix/
readme                                          100%    21     4.9KB/s   00:00
vincent@Wenbins-MacBook-Pro ~ %
```

- Download

```
[vincent@Wenbins-MacBook-Pro ~ % ls
Applications Downloads    Movies        Pictures      gDrive
Desktop        Dropbox      Music         Public        opt
Documents      Library      OneDrive      Zotero        readme
[vincent@Wenbins-MacBook-Pro ~ % scp wbguo@hoffman2.idre.ucla.edu:qcbio-intro2Unix/day1/test .
test                                            100%    34     9.6KB/s   00:00
[vincent@Wenbins-MacBook-Pro ~ % ls
Applications Downloads    Movies        Pictures      gDrive        test
Desktop        Dropbox      Music         Public        opt
Documents      Library      OneDrive      Zotero        readme
[vincent@Wenbins-MacBook-Pro ~ % cat test
hello world
Command line is cool!
vincent@Wenbins-MacBook-Pro ~ %
```

Download & upload (`sftp`)

- sftp: (ssh file transfer protocol)
- Syntax (from local computer terminal):
  - `sftp username@hoffman2.idre.ucla.edu`

- List remote files: `ls`

```
sftp> ls
R                apps              iproject
perl5            project-collab    project-mcdb
project-msehl    qcbio-intro2Unix  run_kde_sim.sh
test             test.RDS          ttest
sftp>
```

- Change remote directory: `cd`

```
sftp> pwd
Remote working directory: /u/home/w/wbguo
sftp> cd qcbio-intro2Unix
sftp> pwd
Remote working directory: /u/home/w/wbguo/qcbio-intro2Unix
```

- List local files: `lls`

```
sftp> lls
Applications    Dropbox     OneDrive    gDrive
Desktop         Library     Pictures    opt
Documents       Movies      Public      readme
Downloads       Music       Zotero      test
sftp>
```

- Change local directory: `lcd`

```
sftp> lpwd
Local working directory: /Users/vincent
sftp> lcd Downloads
sftp> lpwd
Local working directory: /Users/vincent/Downloads
```

Download & upload (`sftp`)

- sftp: (ssh file transfer protocol)
- Syntax (from local computer terminal):
    - `sftp username@hoffman2.idre.ucla.edu`

- Download file to local computer: `get <file name>`

```
sftp> get README.md
Fetching /u/home/w/wbguo/qcbio-intro2Unix/README.md to README.md
/u/home/w/wbguo/qcbio-intro2Unix 100%   19      2.3KB/s   00:00
sftp> lls
Applications    Library         Public          readme
Desktop         Movies          README.md       test
Documents       Music           Zotero
Downloads       OneDrive        gDrive
Dropbox         Pictures        opt
```

- Upload file to remote server: `put <file name>`

```
sftp> ls
README.md  day1        readme      tmp
sftp> put test
Uploading test to /u/home/w/wbguo/qcbio-intro2Unix/test
test                                100%   34      7.0KB/s   00:00
sftp> ls
README.md  day1        readme      test        tmp
sftp> bye
vincent@Wenbins-MacBook-Pro ~ %
```

- Terminate sftp connection: bye

Download & upload (GUI tools option)

Windows user

- MobaXterm

Other tools:
- WinSCP
- FileZilla
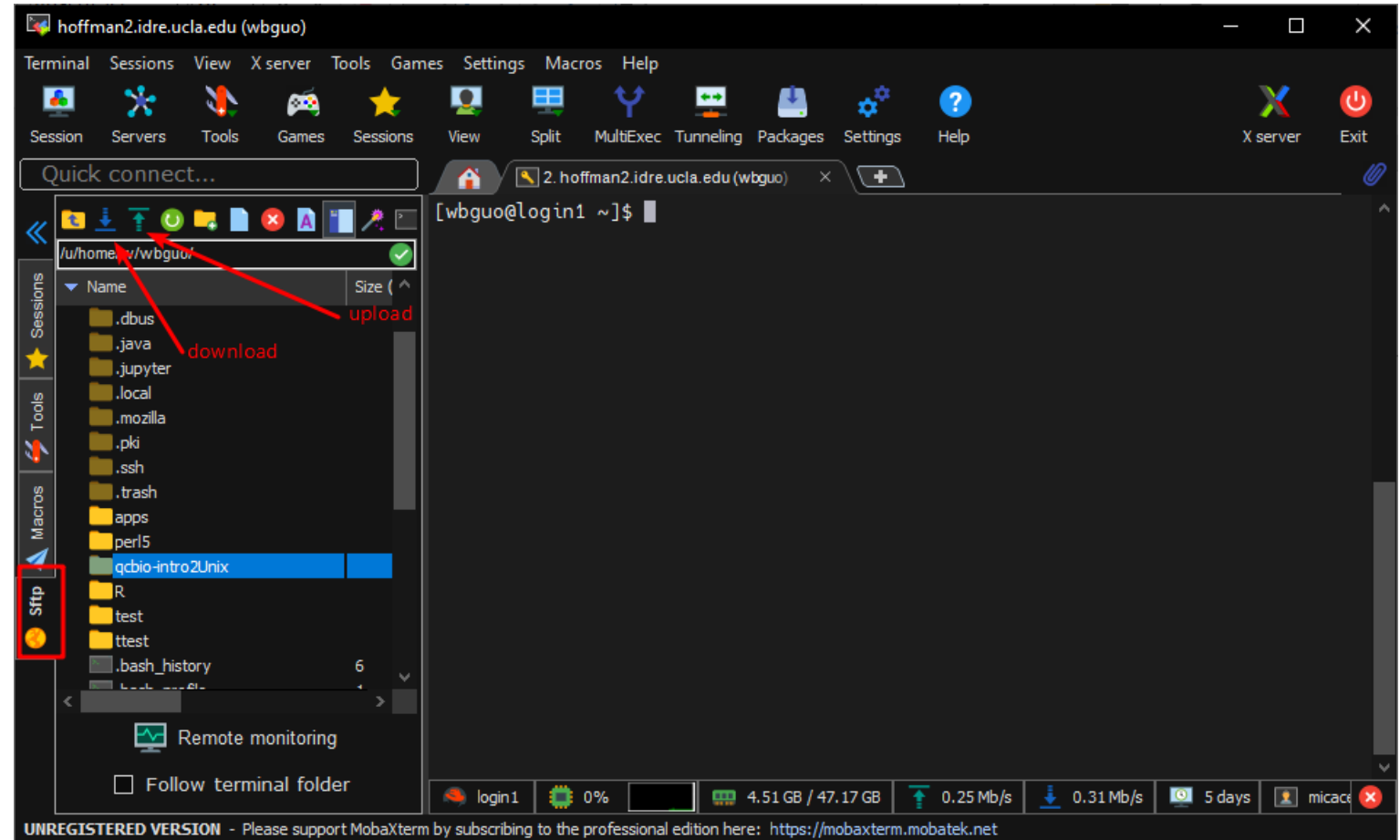
Download & upload (GUI tools option)
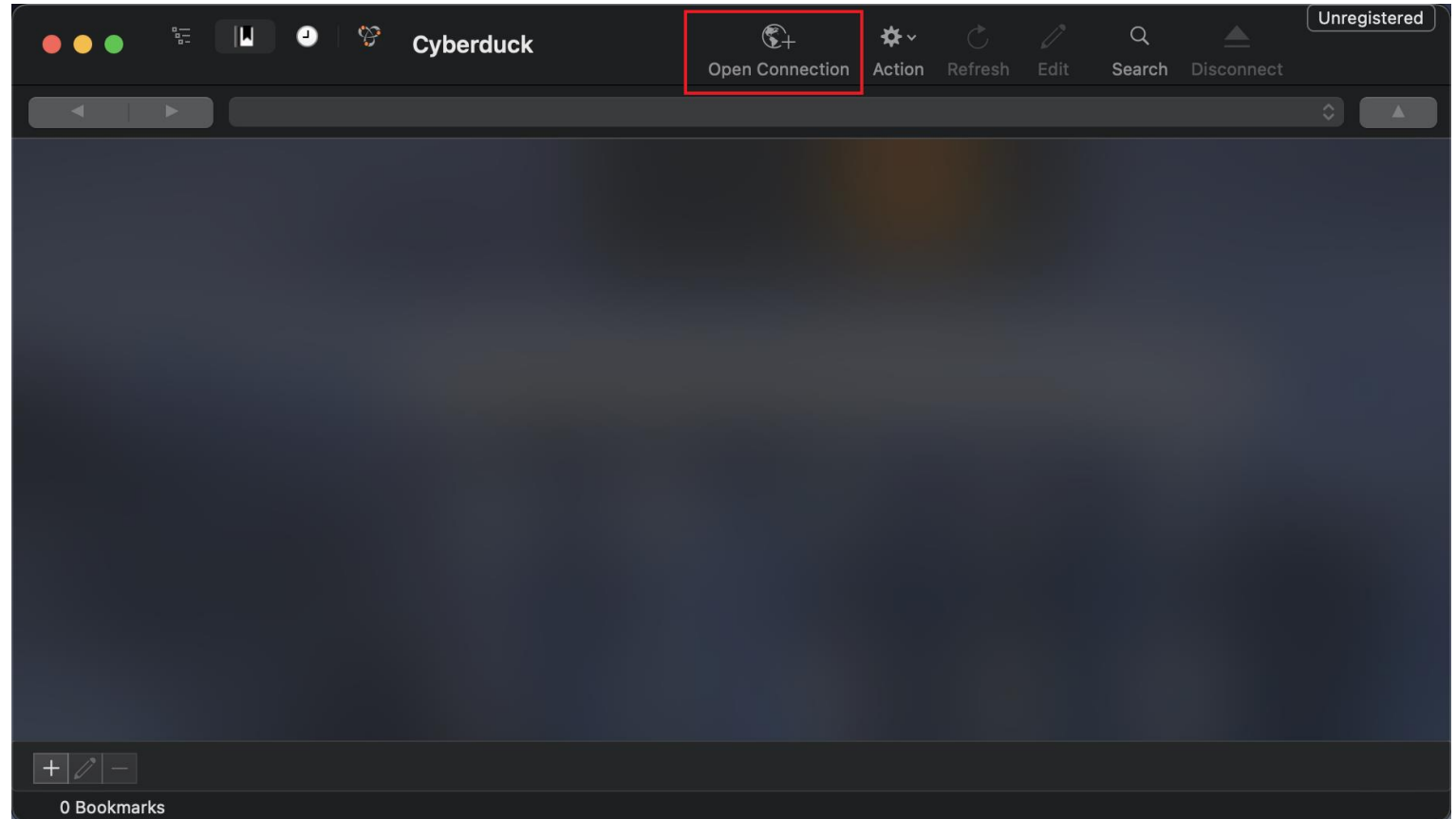
macOS user

- Cyberduck

Other tools:
- FileZilla

Download & upload (GUI tools option)

macOS user

• Cyberduck

Other tools:
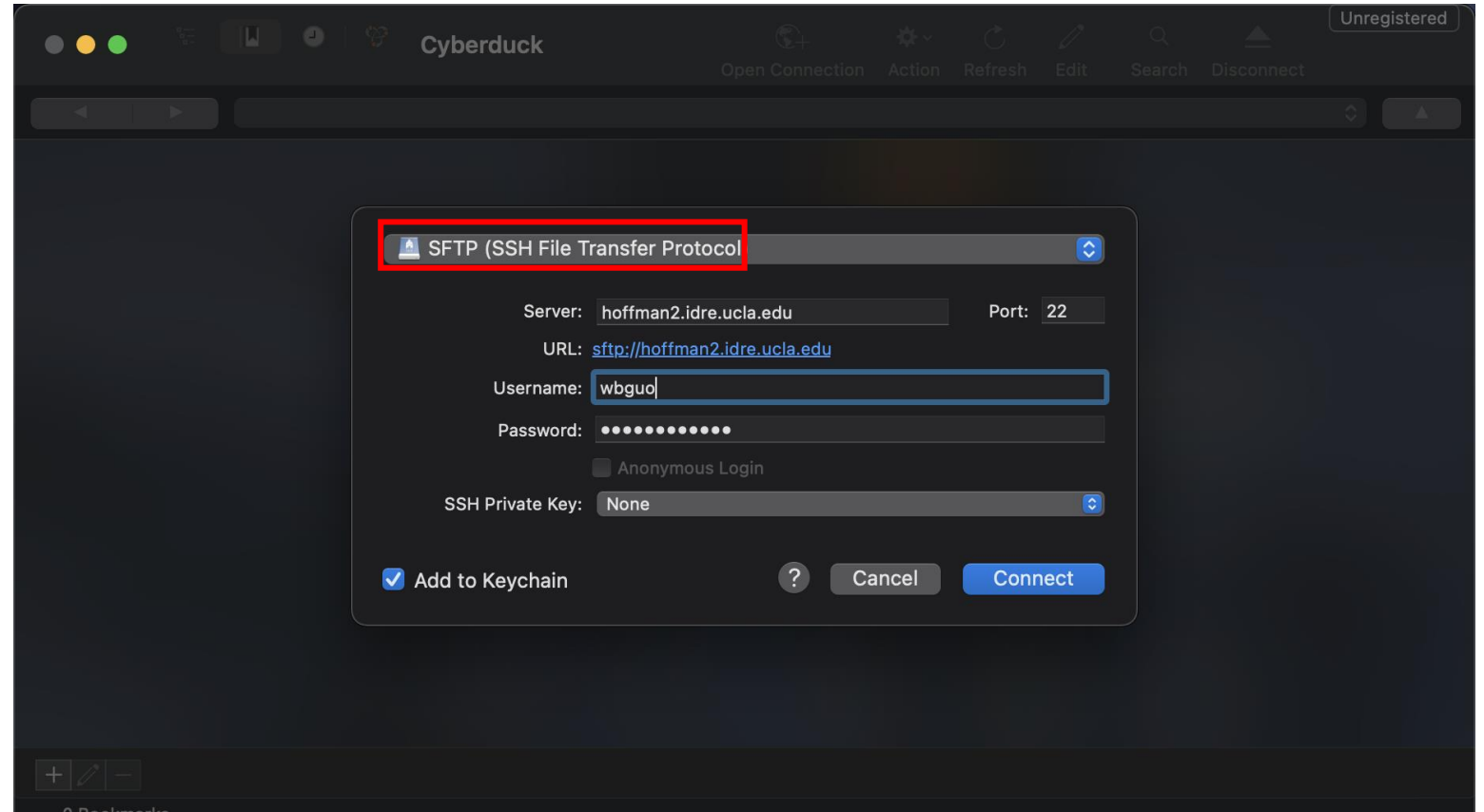
•   FileZilla
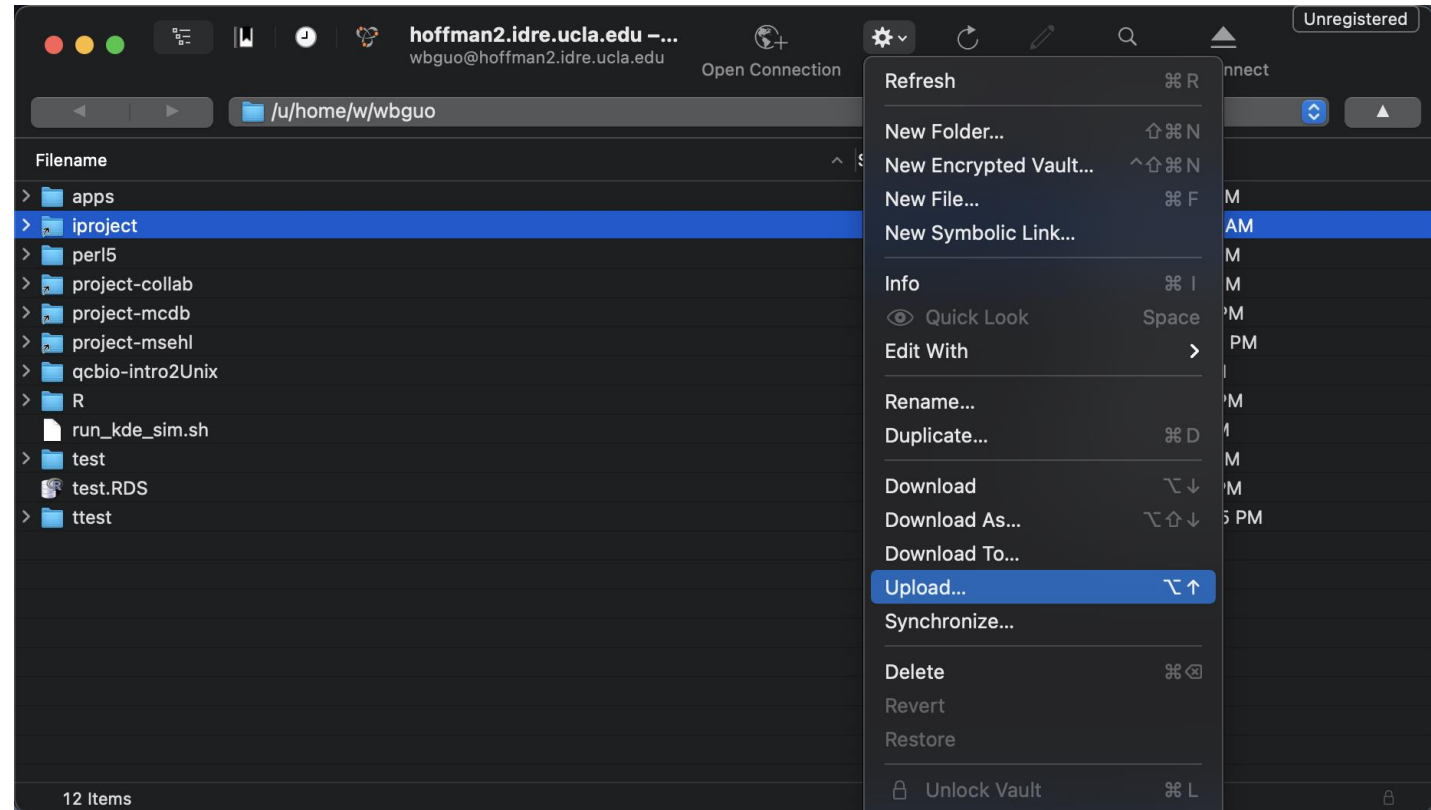
Download & upload (GUI tools option)

macOS user

- Cyberduck

Other tools:
- FileZilla

Practice

- Compress the day1 folder into a `tar.gz` file

- Transfer the file from hoffman2 to your local computer using scp and an GUI application

Close connection to hoffman2

- exit: close ssh connection

Cannot remember so many commands?

- No problem!

- **--help**
  - Usage: `command --help`

Cannot remember so many commands?

- No problem!

- `--help`
  - Usage: `command --help`

- man: manual
  - Usage: `man command`

Use up/down arrow to scroll line by line
Use PgDn/PgUp to scroll page by page
Press q for exit

```
LS(1)                           User Commands                          LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about the FILEs (the current directory by default).  Sort
       entries alphabetically if none of -cftuvSUX nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              scale sizes by SIZE  before  printing  them;  e.g.,  '--block-size=M'
              prints sizes in units of 1,048,576 bytes; see SIZE format below

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with -lt: sort by, and show, ctime (time of last modification of file
              status information); with -l: show ctime and sort by name; otherwise:
              sort by ctime, newest first

       -C     list entries by columns

       --color[=WHEN]
 Manual page ls(1) line 1 (press h for help or q to quit)
```

Where to get help?

- https://www.google.com

- https://stackoverflow.com

- https://unix.stackexchange.com

- https://www.tutorialspoint.com/unix/shell_scripting.htm

Homework

- Read about ssh alias. Create an alias for hoffman2
  https://ostechnix.com/how-to-create-ssh-alias-in-linux/

- Read about ssh login without password. Setup login without password for hoffman2
  http://www.linuxproblem.org/art_9.html

Q&A

[Google doc](#)