UCLA QCBio Collaboratory

Introduction to Unix command line & Shell programming

Wenbin Guo

Bioinformatics IDP, UCLA

wbguo@ucla.edu

2022 Spring

UCLA Bioinformatics

Thank Dr. Diane Lefaudeux for base slides

Agenda

- Day 0: 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

Day 0: Preparation

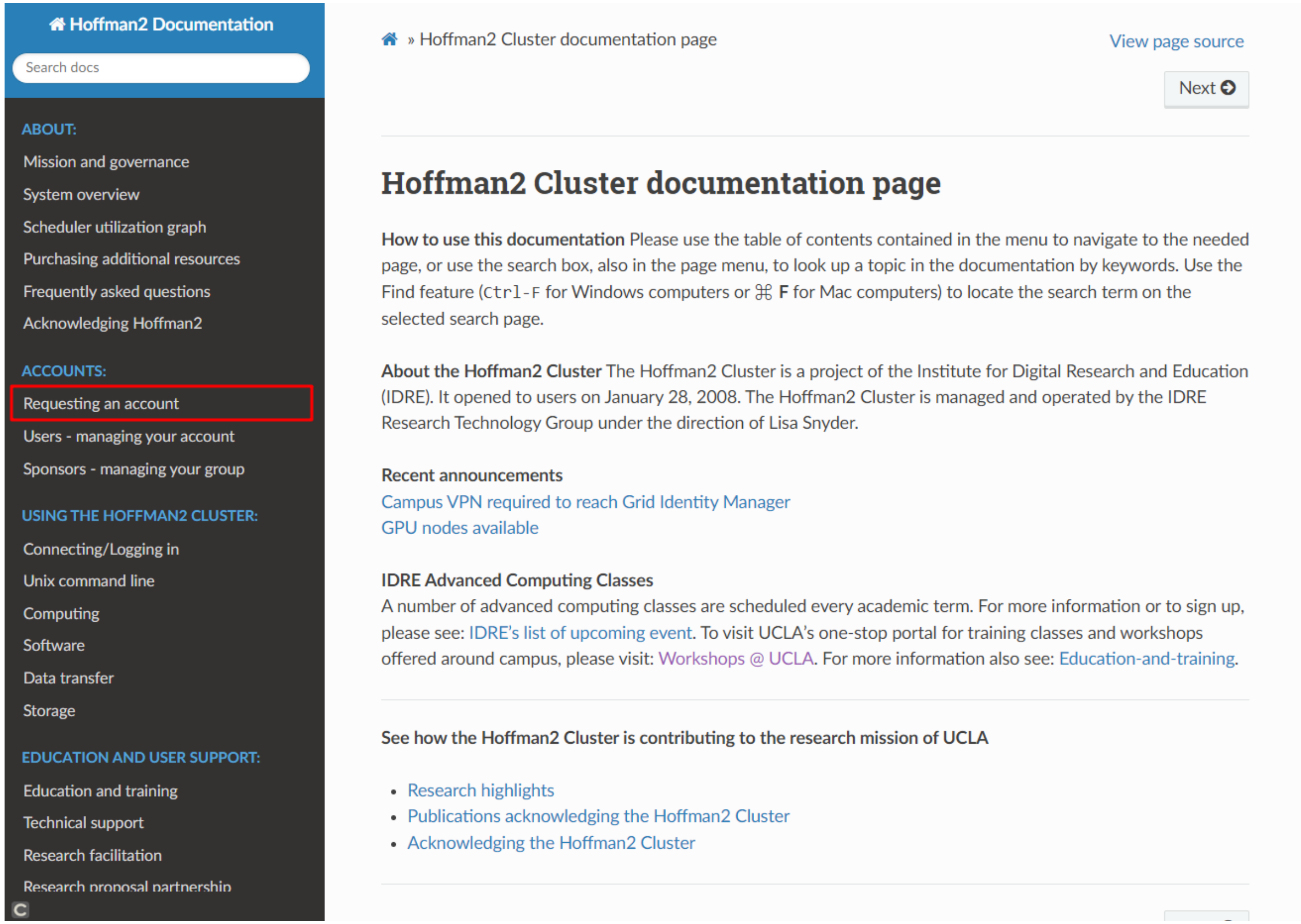
Wenbin Guo

Bioinformatics IDP, UCLA

wbguo@ucla.edu

Step 1: Register a hoffman2 account

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



Hoffman2 Cluster documentation page

How to use this documentation Please use the table of contents contained in the menu to navigate to the needed page, or use the search box, also in the page menu, to look up a topic in the documentation by keywords. Use the Find feature (`Ctrl-F` for Windows computers or ⌘ **F** for Mac computers) to locate the search term on the selected search page.

About the Hoffman2 Cluster The Hoffman2 Cluster is a project of the Institute for Digital Research and Education (IDRE). It opened to users on January 28, 2008. The Hoffman2 Cluster is managed and operated by the IDRE Research Technology Group under the direction of Lisa Snyder.

Recent announcements
Campus VPN required to reach Grid Identity Manager
GPU nodes available

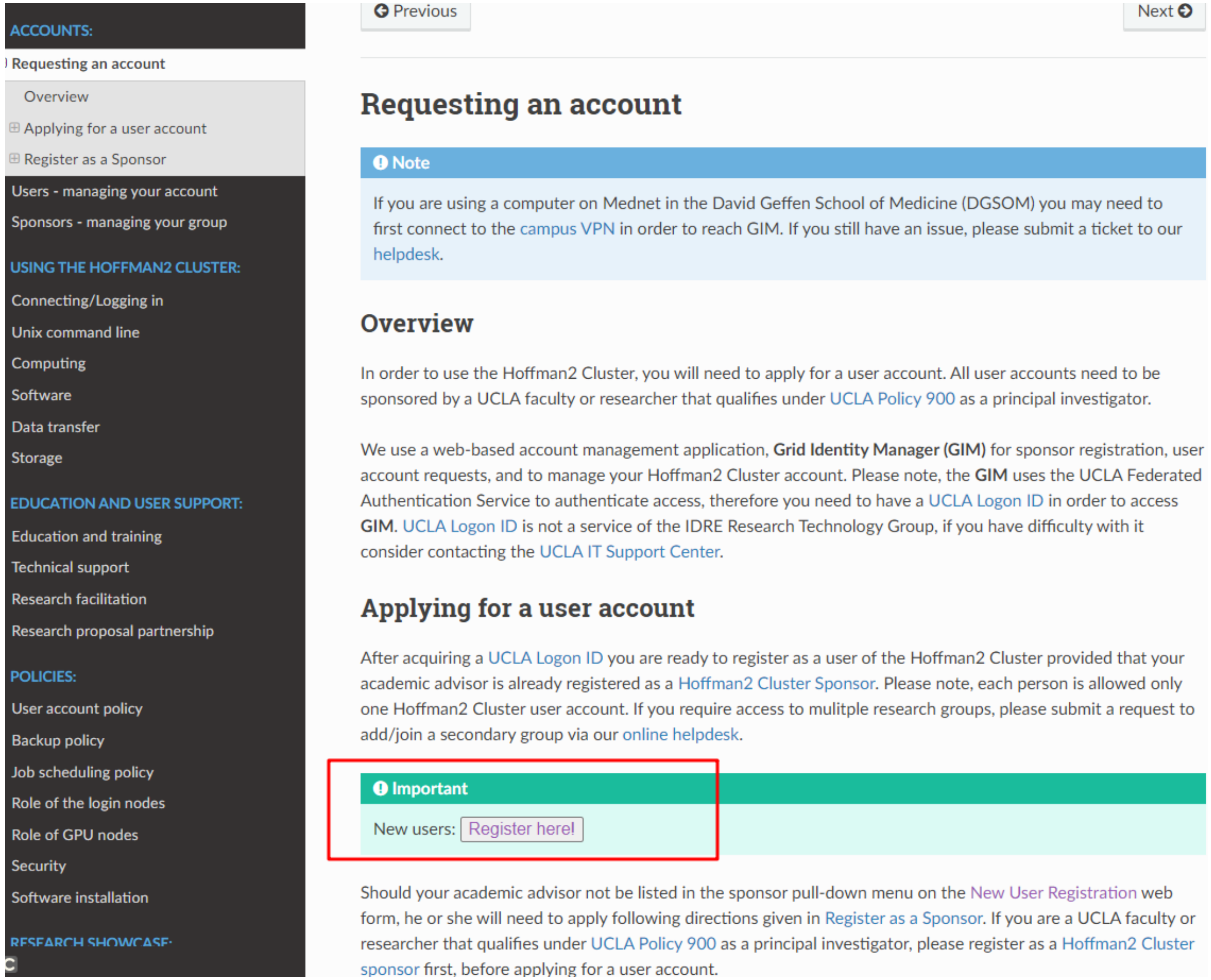
IDRE Advanced Computing Classes
A number of advanced computing classes are scheduled every academic term. For more information or to sign up, please see: IDRE's list of upcoming event. To visit UCLA's one-stop portal for training classes and workshops offered around campus, please visit: Workshops @ UCLA. For more information also see: Education-and-training.

See how the Hoffman2 Cluster is contributing to the research mission of UCLA

- Research highlights
- Publications acknowledging the Hoffman2 Cluster
- Acknowledging the Hoffman2 Cluster

Step 1: Register a hoffman2 account

- Click Register here!



Requesting an account

Note

If you are using a computer on Mednet in the David Geffen School of Medicine (DGSOM) you may need to first connect to the campus VPN in order to reach GIM. If you still have an issue, please submit a ticket to our helpdesk.

Overview

In order to use the Hoffman2 Cluster, you will need to apply for a user account. All user accounts need to be sponsored by a UCLA faculty or researcher that qualifies under UCLA Policy 900 as a principal investigator.

We use a web-based account management application, **Grid Identity Manager (GIM)** for sponsor registration, user account requests, and to manage your Hoffman2 Cluster account. Please note, the **GIM** uses the UCLA Federated Authentication Service to authenticate access, therefore you need to have a UCLA Logon ID in order to access **GIM**. UCLA Logon ID is not a service of the IDRE Research Technology Group, if you have difficulty with it consider contacting the UCLA IT Support Center.

Applying for a user account

After acquiring a UCLA Logon ID you are ready to register as a user of the Hoffman2 Cluster provided that your academic advisor is already registered as a Hoffman2 Cluster Sponsor. Please note, each person is allowed only one Hoffman2 Cluster user account. If you require access to mulitple research groups, please submit a request to add/join a secondary group via our online helpdesk.

Important

New users: Register here!

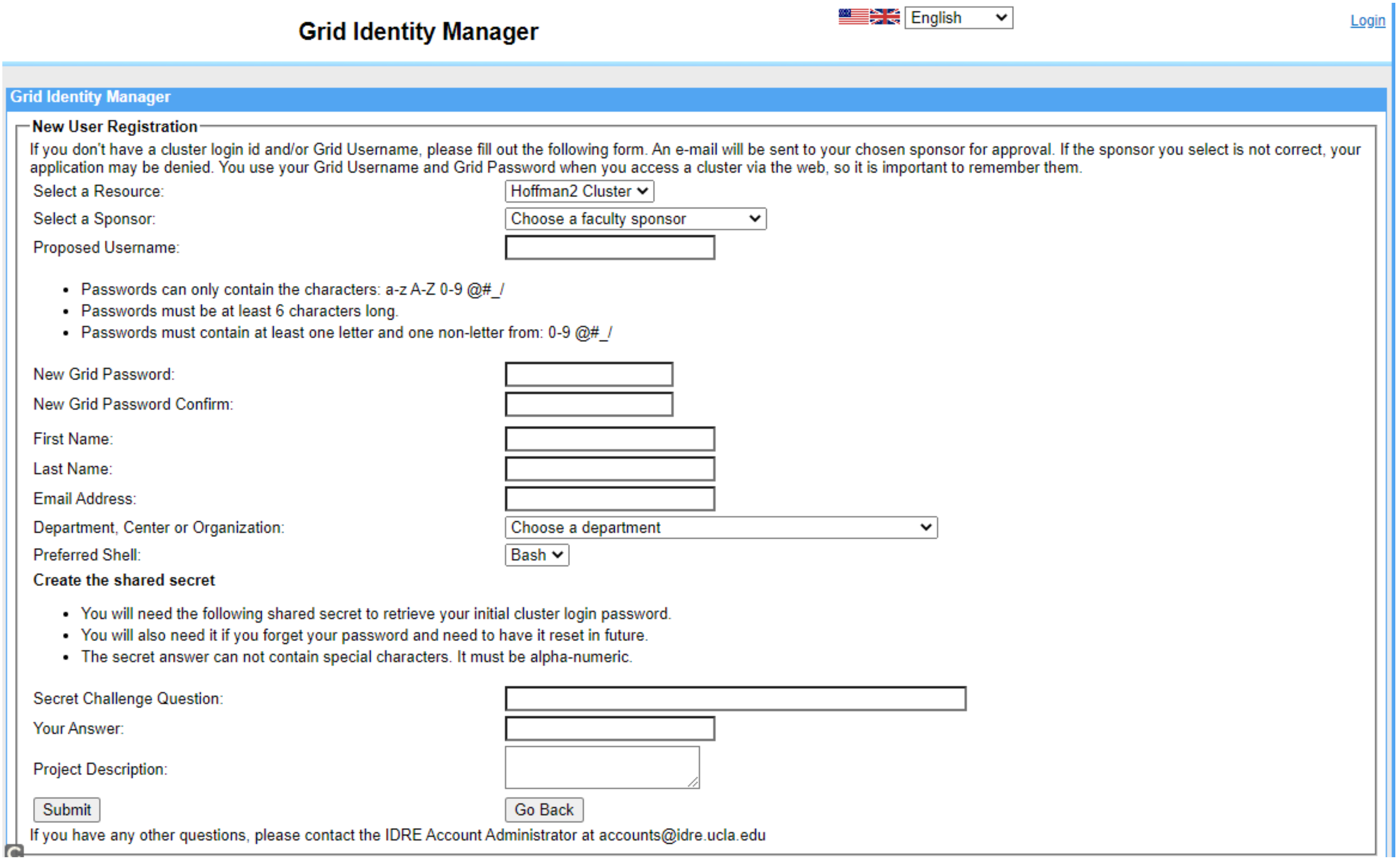
Should your academic advisor not be listed in the sponsor pull-down menu on the New User Registration web form, he or she will need to apply following directions given in Register as a Sponsor. If you are a UCLA faculty or researcher that qualifies under UCLA Policy 900 as a principal investigator, please register as a Hoffman2 Cluster sponsor first, before applying for a user account.

(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)



Grid Identity Manager

Grid Identity Manager

New User Registration

If you don't have a cluster login id and/or Grid Username, please fill out the following form. An e-mail will be sent to your chosen sponsor for approval. If the sponsor you select is not correct, your application may be denied. You use your Grid Username and Grid Password when you access a cluster via the web, so it is important to remember them.

Select a Resource:	Hoffman2 Cluster
Select a Sponsor:	Choose a faculty sponsor
Proposed Username:	

- Passwords can only contain the characters: a-z A-Z 0-9 @#_/
- Passwords must be at least 6 characters long.
- Passwords must contain at least one letter and one non-letter from: 0-9 @#_/

New Grid Password:	
New Grid Password Confirm:	
First Name:	
Last Name:	
Email Address:	
Department, Center or Organization:	Choose a department
Preferred Shell:	Bash

Create the shared secret

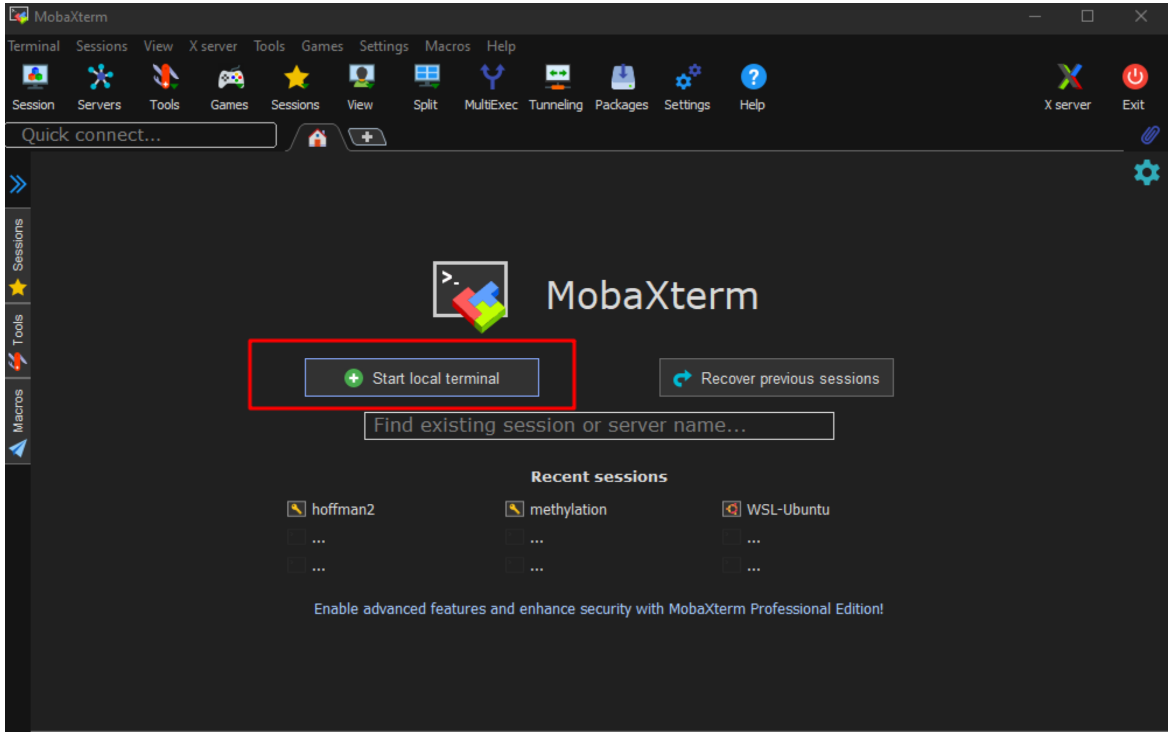
- You will need the following shared secret to retrieve your initial cluster login password.
- You will also need it if you forget your password and need to have it reset in future.
- The secret answer can not contain special characters. It must be alpha-numeric.

Secret Challenge Question:	
Your Answer:	
Project Description:	

If you have any other questions, please contact the IDRE Account Administrator at accounts@idre.ucla.edu

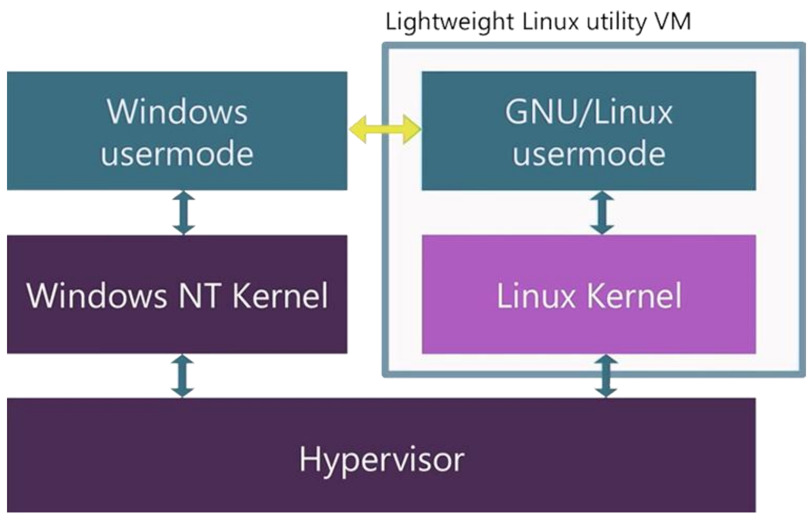
Step 2: Get your computer ready!

- 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


MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
Sessions
Tools
Macros
MobaXterm
Start local terminal
Recover previous sessions
Find existing session or server name...
Recent sessions
hoffman2
methylation
WSL-Ubuntu
...
Enable advanced features and enhance security with MobaXterm Professional Edition!

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



Lightweight Linux utility VM

Windows usermode

GNU/Linux usermode

Windows NT Kernel

Linux Kernel

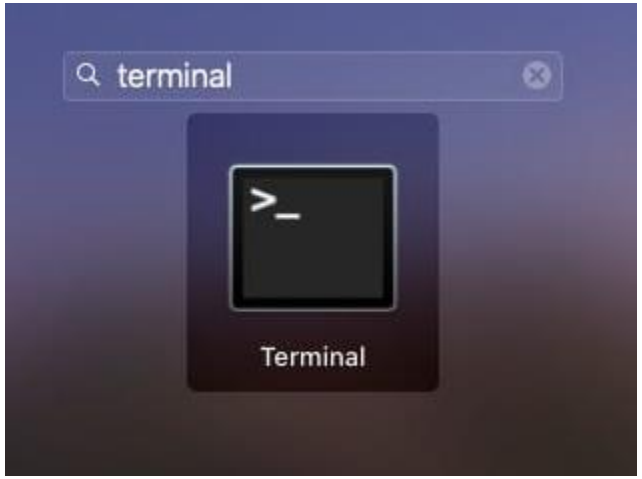
Hypervisor

Step 2: Get your computer ready!

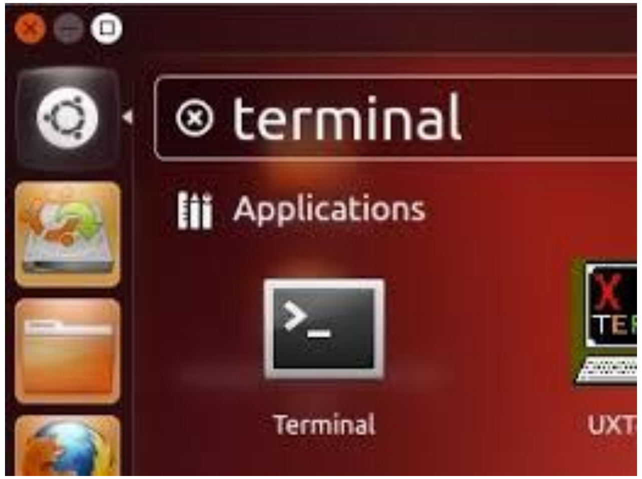
- If you are a Windows user
  - Option 1: Install MobaXterm
    - Download the free software https://mobaxterm.mobatek.net/download.html
    - Install and start local terminal
  - Option 2: Install Windows Subsystem Linux (WSL)
    - Follow this official instructions
    - Open the terminal
  - Option 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

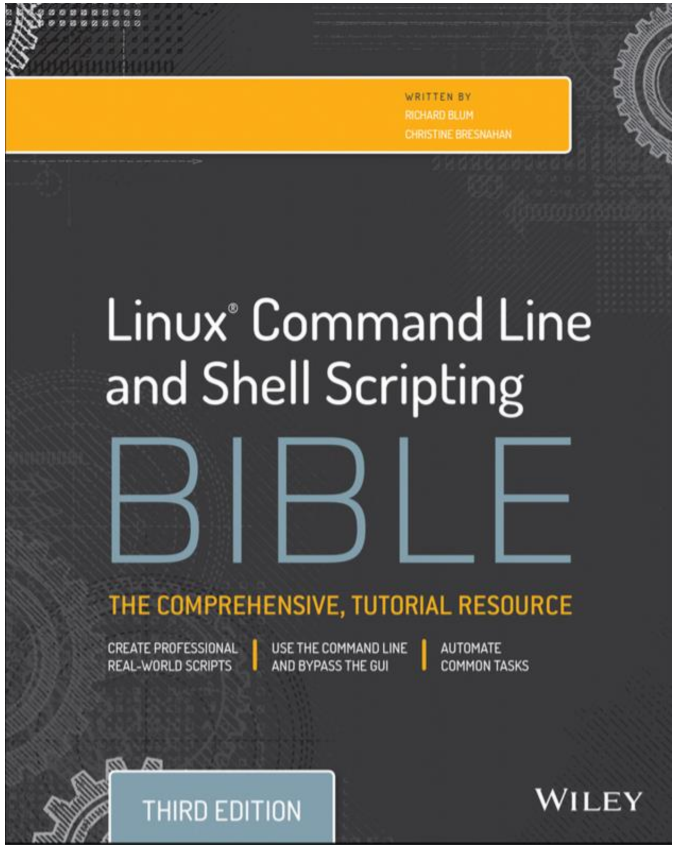


ubuntu

Questions and references

- Write down questions at this Google doc

- Reference book is freely available online



WRITTEN BY
RICHARD BLUM
CHRISTINE BRESNAHAN

Linux® Command Line and Shell Scripting

BIBLE

THE COMPREHENSIVE, TUTORIAL RESOURCE

CREATE PROFESSIONAL REAL-WORLD SCRIPTS | USE THE COMMAND LINE AND BYPASS THE GUI | AUTOMATE COMMON TASKS

THIRD EDITION

WILEY

UCLA QCBio
Collaboratory

Day 1: Introduction to Unix command line

Wenbin Guo
Bioinformatics IDP, UCLA
wbguo@ucla.edu

2022/03/29

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

https://www.hostinger.com/tutorials/linux-commands

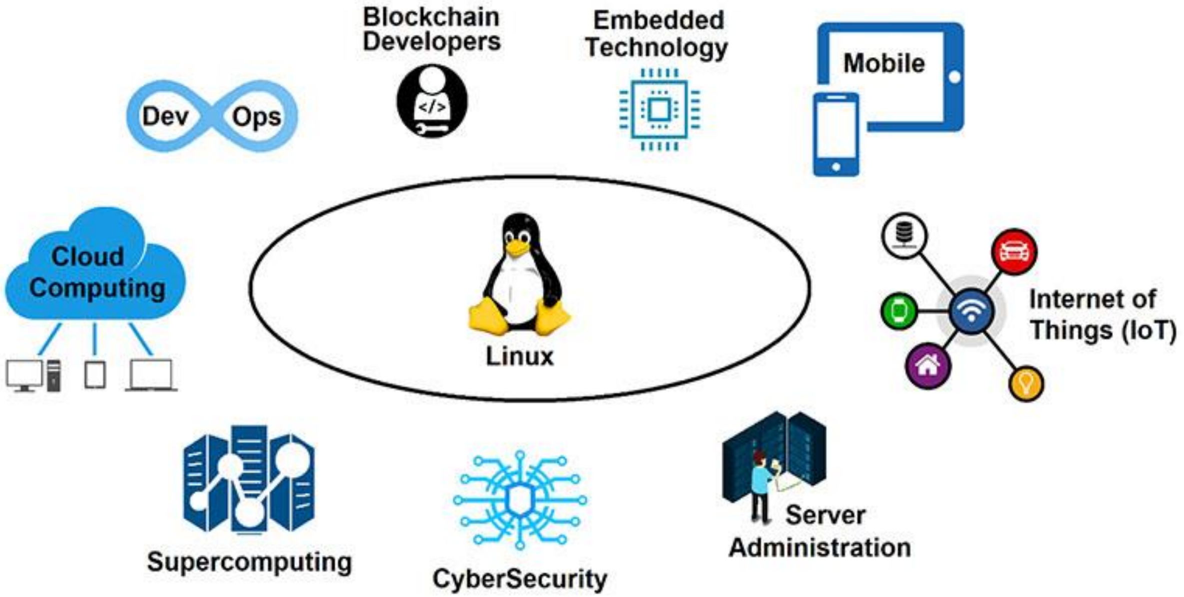
What is Unix/Linux?

- 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

…

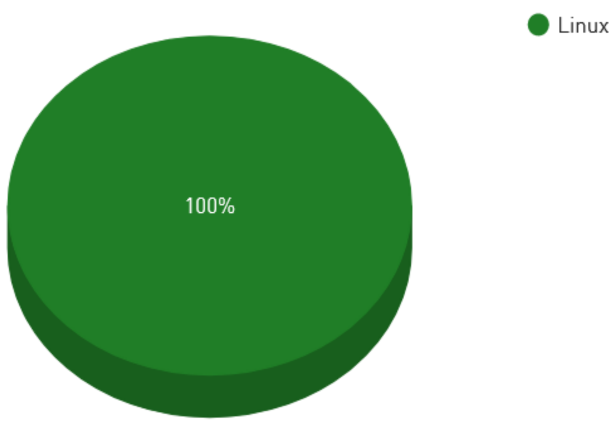


Unix/Linux is widely used in academia and industry!

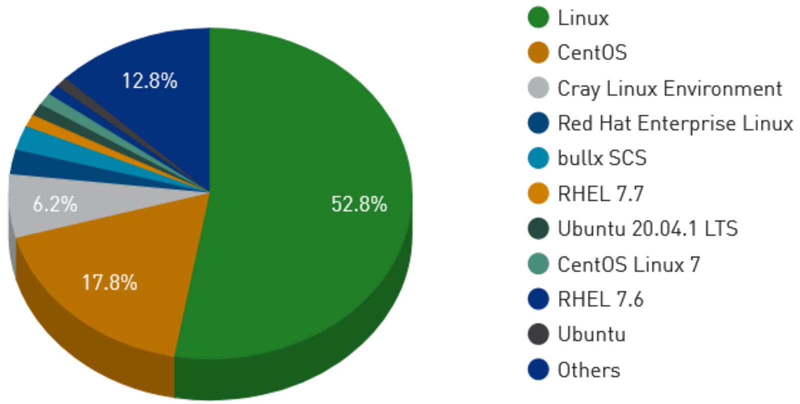
Why Unix/Linux?

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

Operating system Family System Share



Operating System System Share

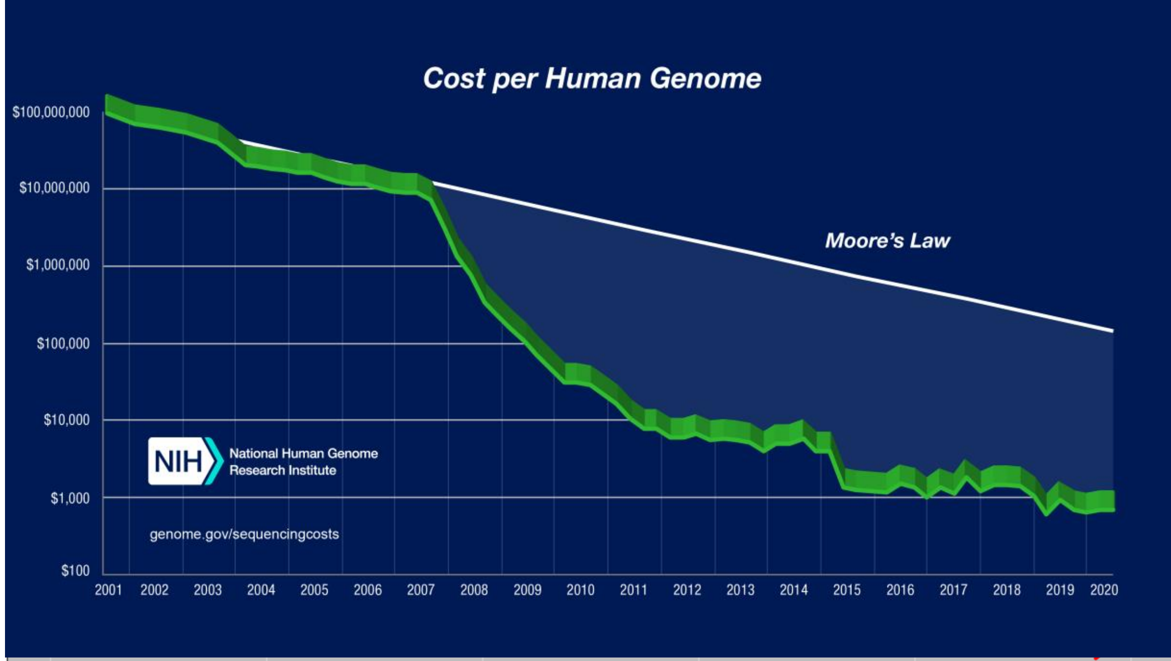


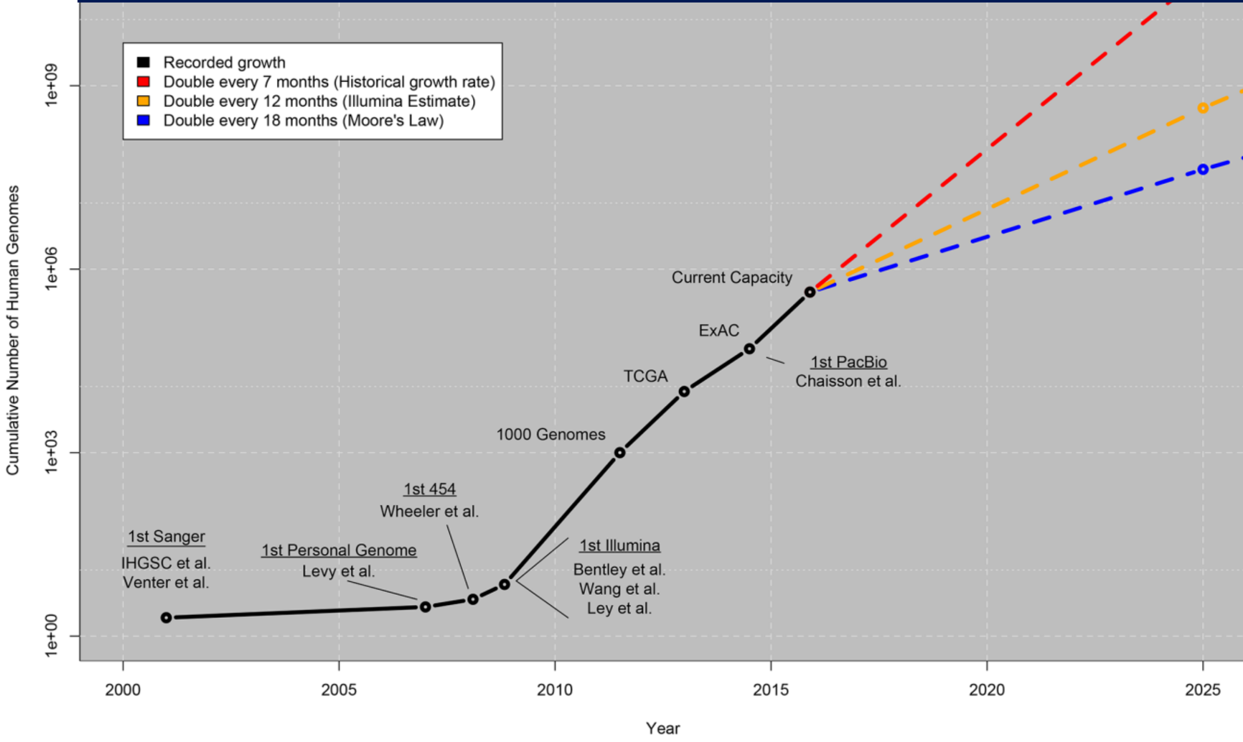
www.top500.org (June 2021)

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



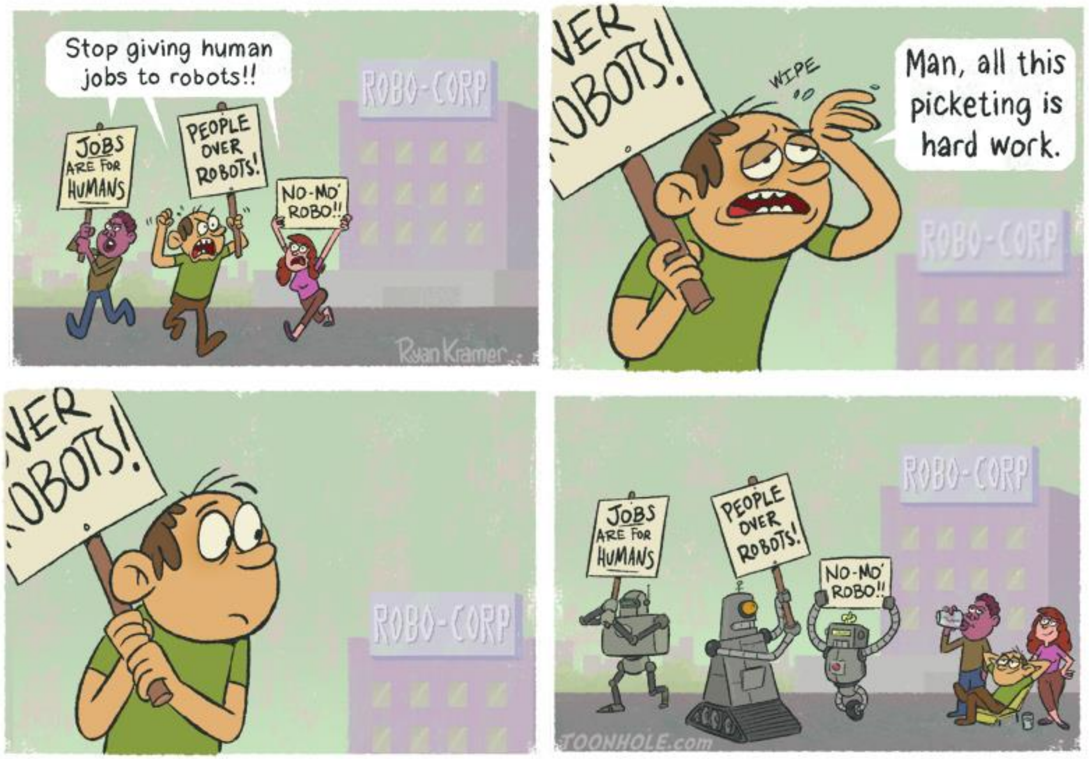


Moore's law: the number of transistors in microchip doubles every 18 months

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

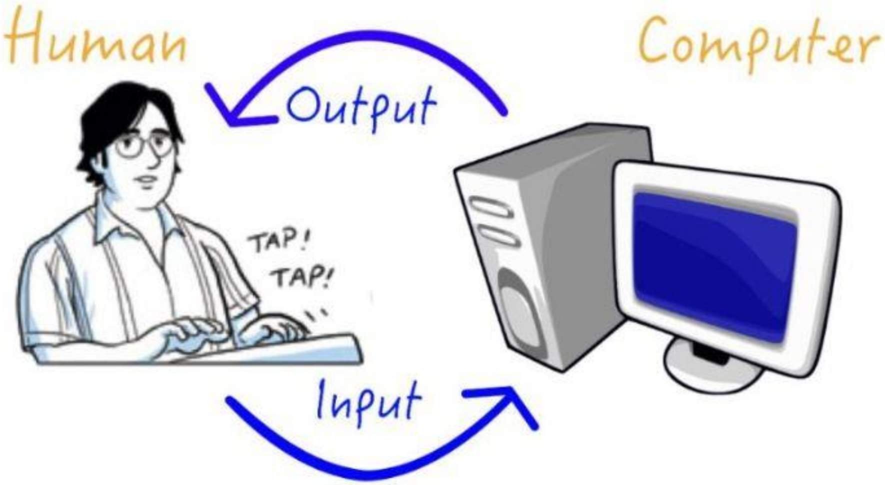


http://joyreactor.com/post/2632813

How to give orders to the machine?

Ideal human-computer interaction

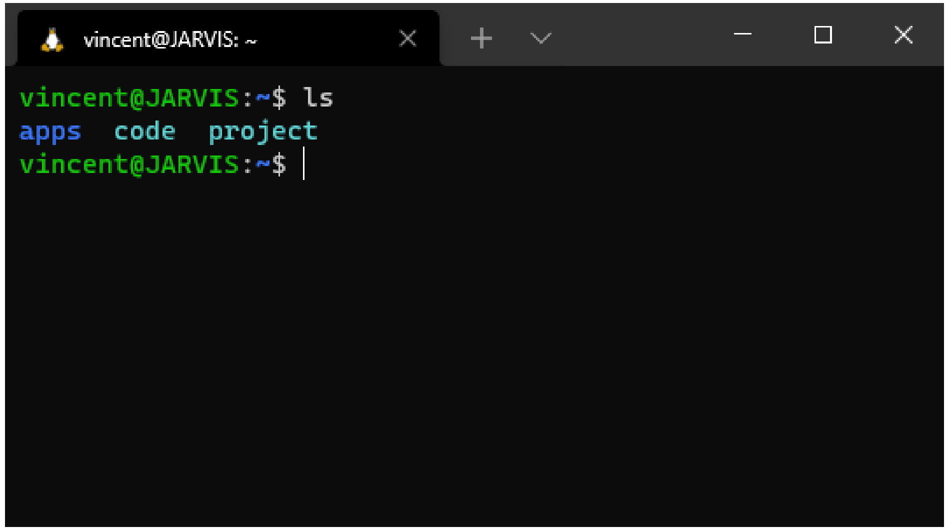
Actual human-computer interaction



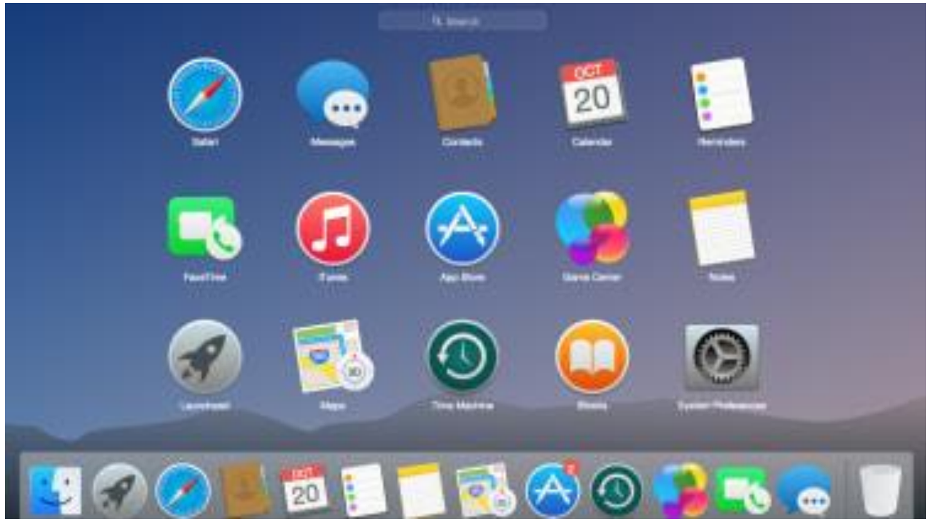
Use Command Line!

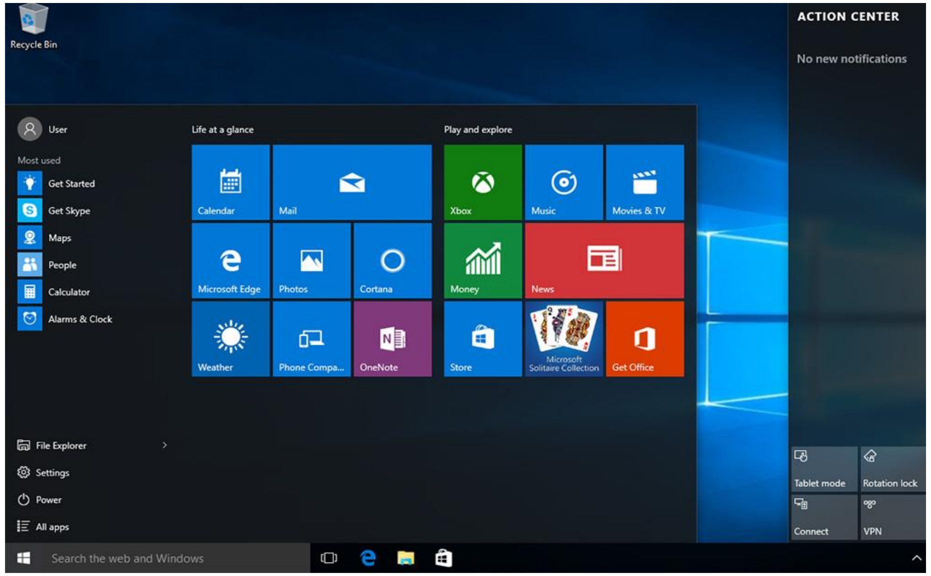
Screencap from Ironman movie;
https://www.hiof.no/iio/itk/forskning/grupper/interaction-design/aktuelt/what-is-a-human-computer-interface-.html

Command Line Interface vs Graphical User Interface

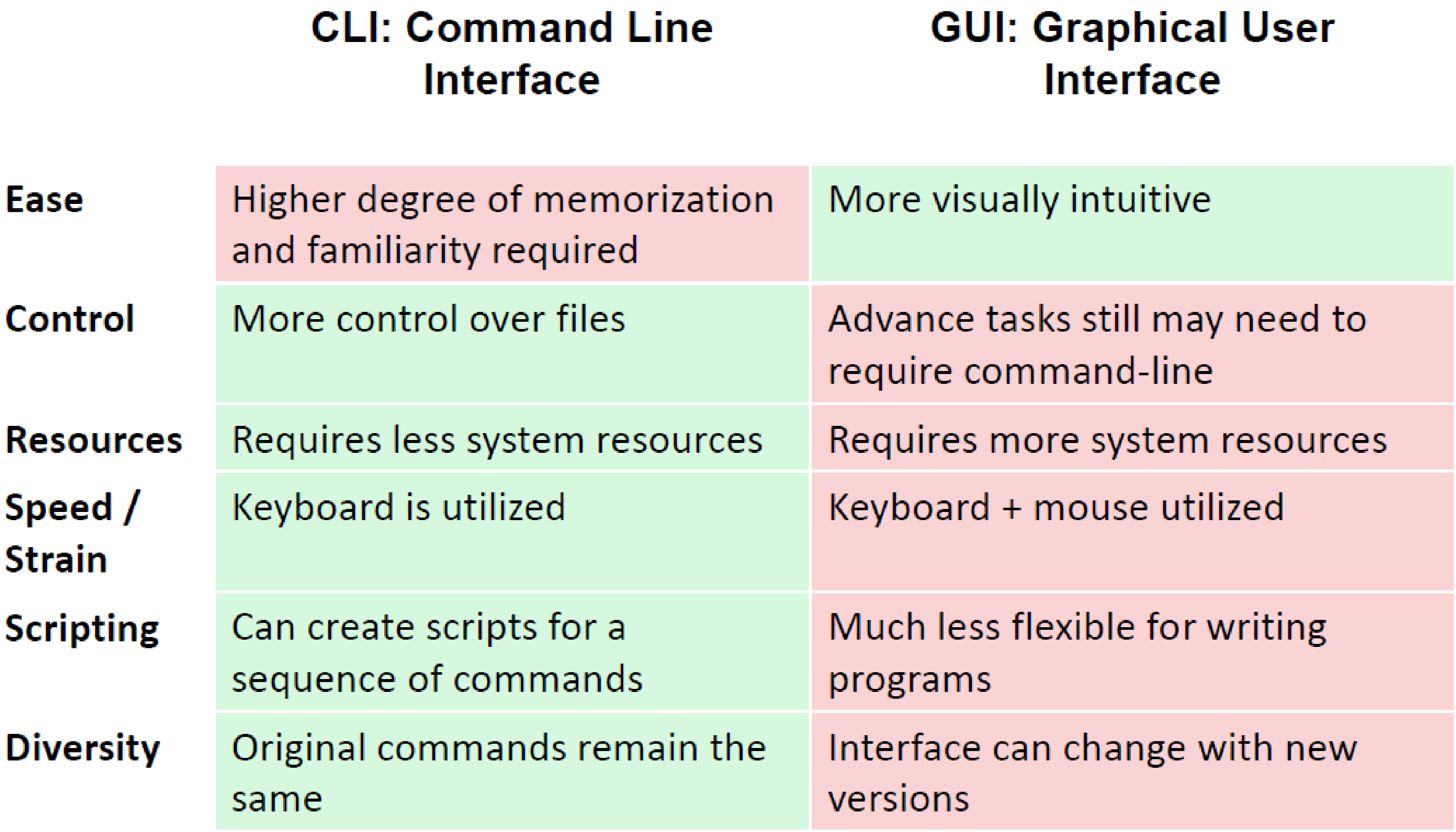


```
vincent@JARVIS:~$ ls
apps  code  project
vincent@JARVIS:~$
```





Command Line Interface vs Graphical User Interface

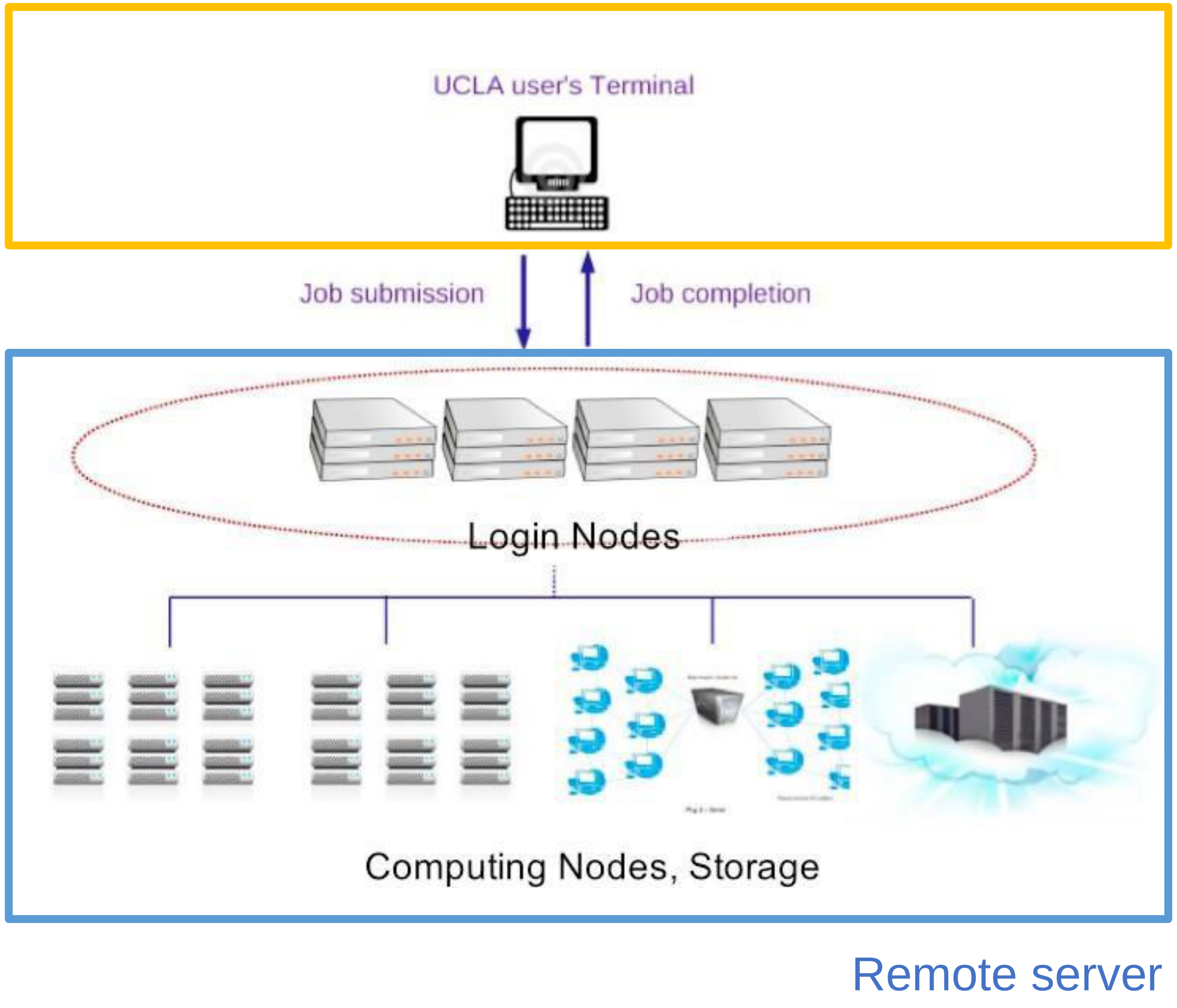


	CLI: Command Line Interface	GUI: Graphical User Interface
Ease	Higher degree of memorization and familiarity required	More visually intuitive
Control	More control over files	Advance tasks still may need to require command-line
Resources	Requires less system resources	Requires more system resources
Speed / 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.



Local computer

UCLA user's Terminal

Job submission

Job completion

Login Nodes

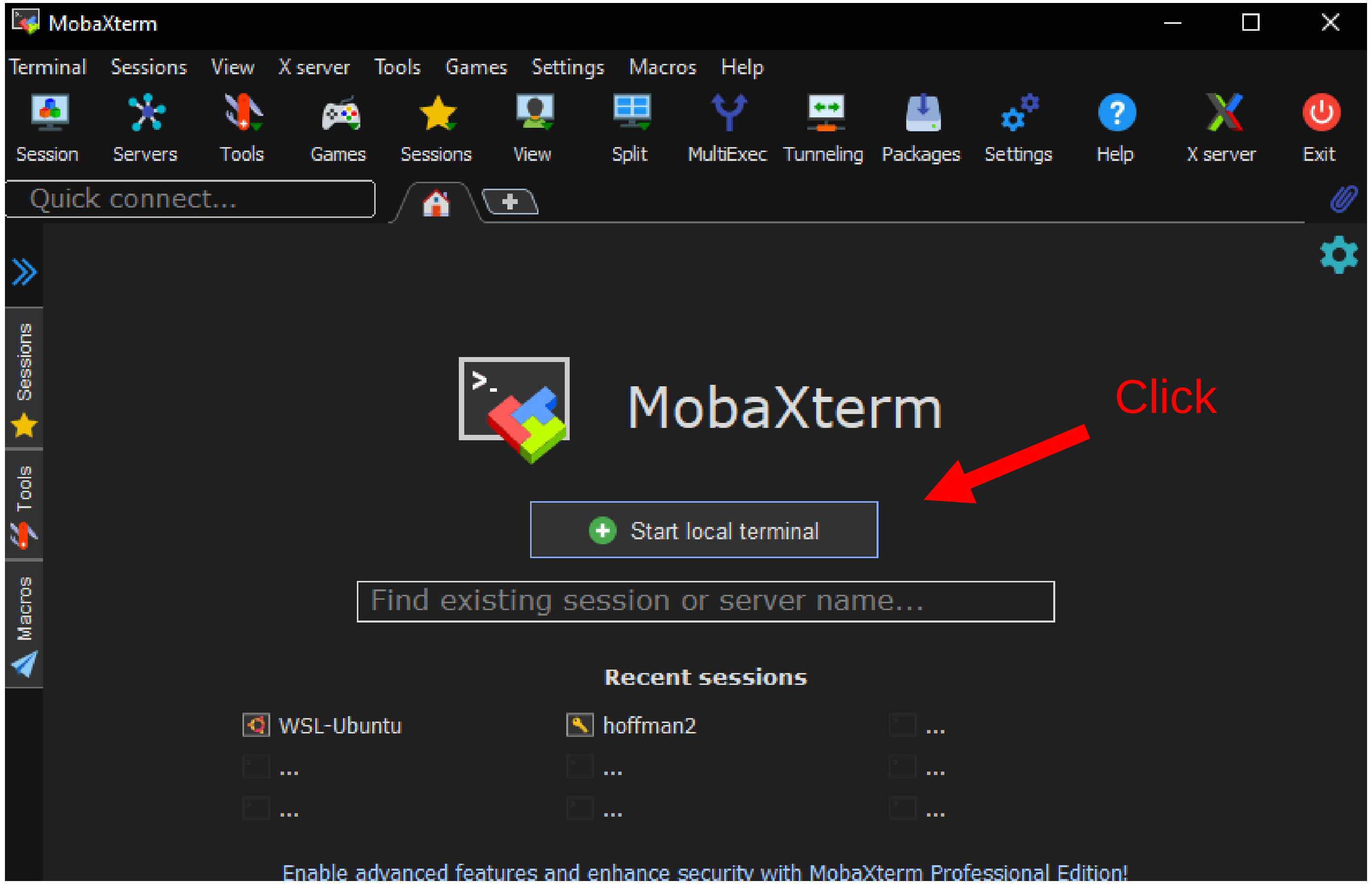
Computing Nodes, Storage

Remote server

Let’s login to hoffman2 from command line interface (CLI)!

Access hoffman2 (Windows user)

Start local terminal:



Enable advanced features and enhance security with MobaXterm Professional Edition!

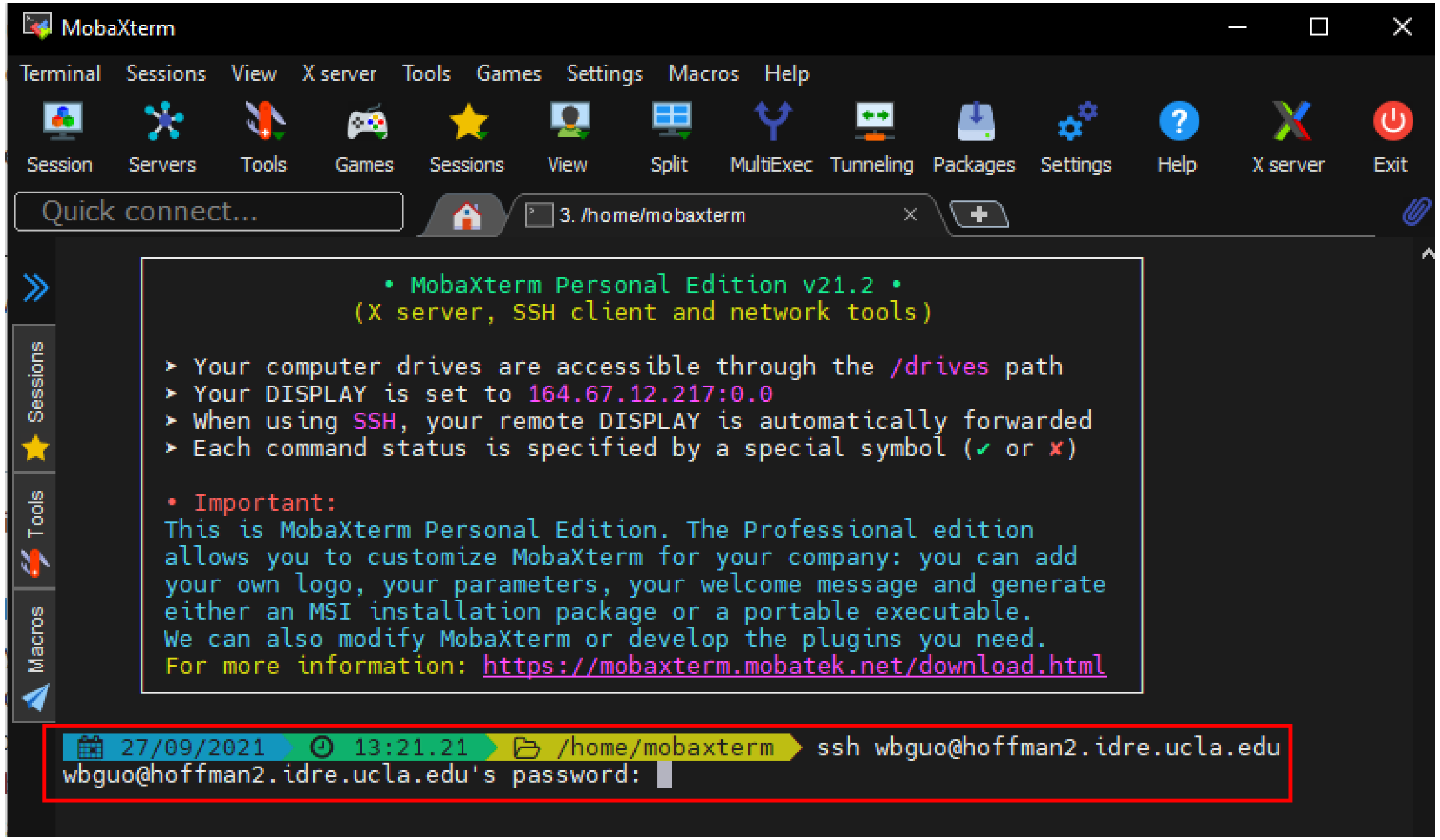
Click

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*)



MobaXterm
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit
Quick connect...
3. /home/mobaxterm
Sessions
Tools
Macros

```
• MobaXterm Personal Edition v21.2 •
(X server, SSH client and network tools)

➤ Your computer drives are accessible through the /drives path
➤ Your DISPLAY is set to 164.67.12.217:0.0
➤ When using SSH, your remote DISPLAY is automatically forwarded
➤ Each command status is specified by a special symbol (✔ or ✘)

• Important:
This is MobaXterm Personal Edition. The Professional edition
allows you to customize MobaXterm for your company: you can add
your own logo, your parameters, your welcome message and generate
either an MSI installation package or a portable executable.
We can also modify MobaXterm or develop the plugins you need.
For more information: https://mobaxterm.mobatek.net/download.html

 27/09/2021  13:21.21  /home/mobaxterm  ssh wbguo@hoffman2.idre.ucla.edu
wbguo@hoffman2.idre.ucla.edu's password:
```

Note:

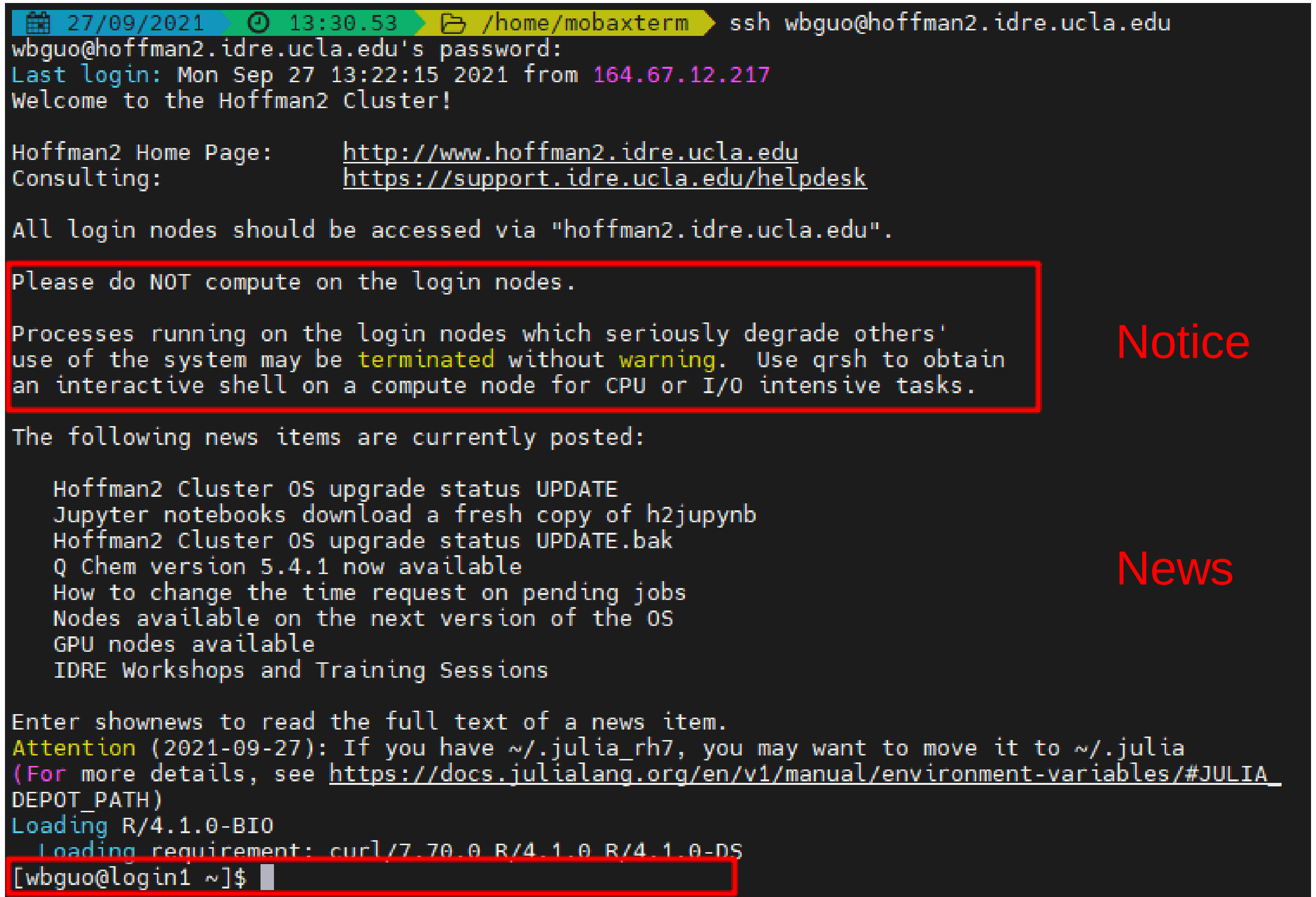
- Command line is case-sensitive
- For security reason, when you type in password, nothing shows up on the screen

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*)



```
 27/09/2021   13:30.53   /home/mobaxterm  ssh wbguo@hoffman2.idre.ucla.edu
wbguo@hoffman2.idre.ucla.edu's password:
Last login: Mon Sep 27 13:22:15 2021 from 164.67.12.217
Welcome to the Hoffman2 Cluster!

Hoffman2 Home Page:     http://www.hoffman2.idre.ucla.edu
Consulting:             https://support.idre.ucla.edu/helpdesk

All login nodes should be accessed via "hoffman2.idre.ucla.edu".

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/#JULIA_
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 ~]$
```

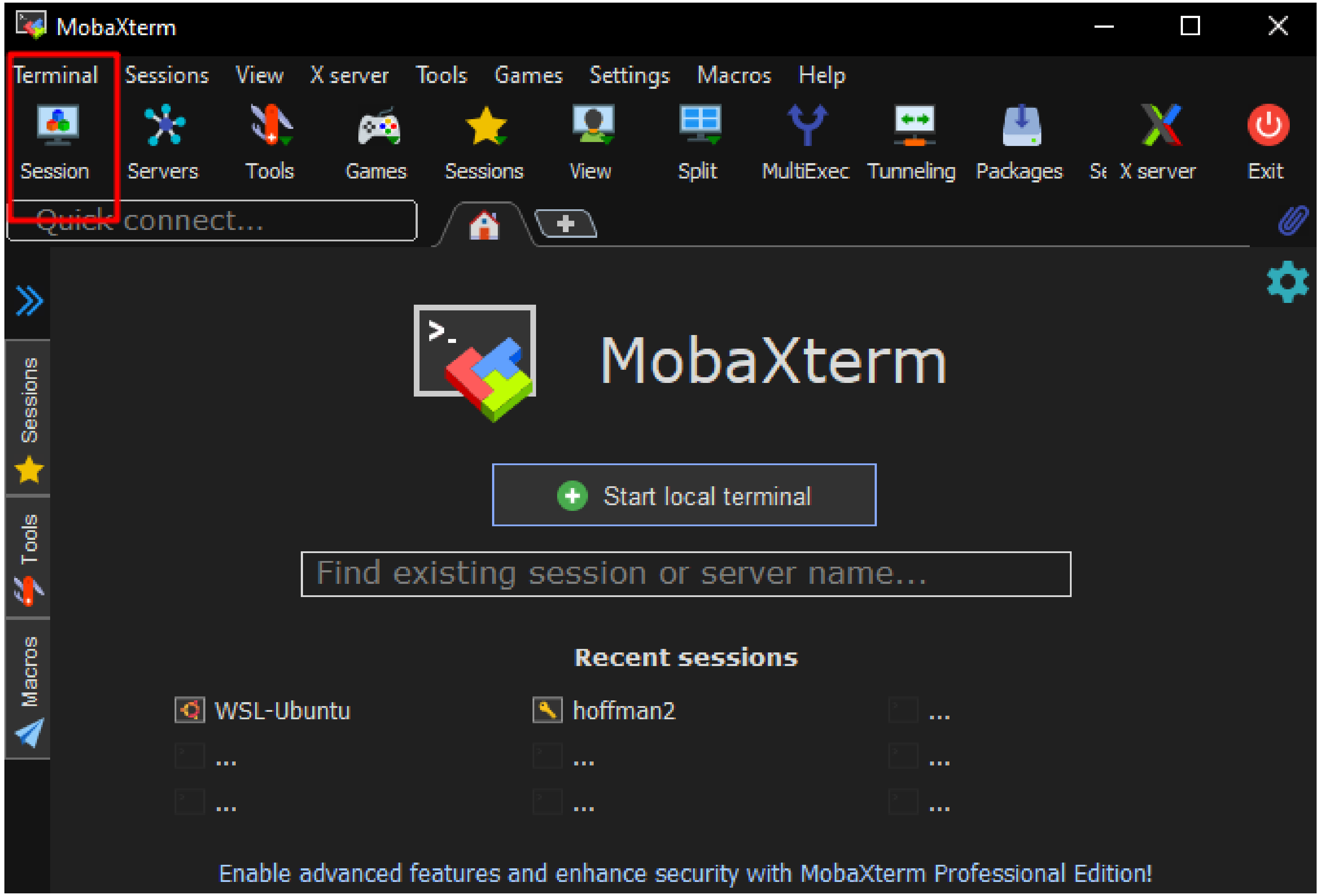
Notice

News

Access hoffman2 (Windows user)

Another option

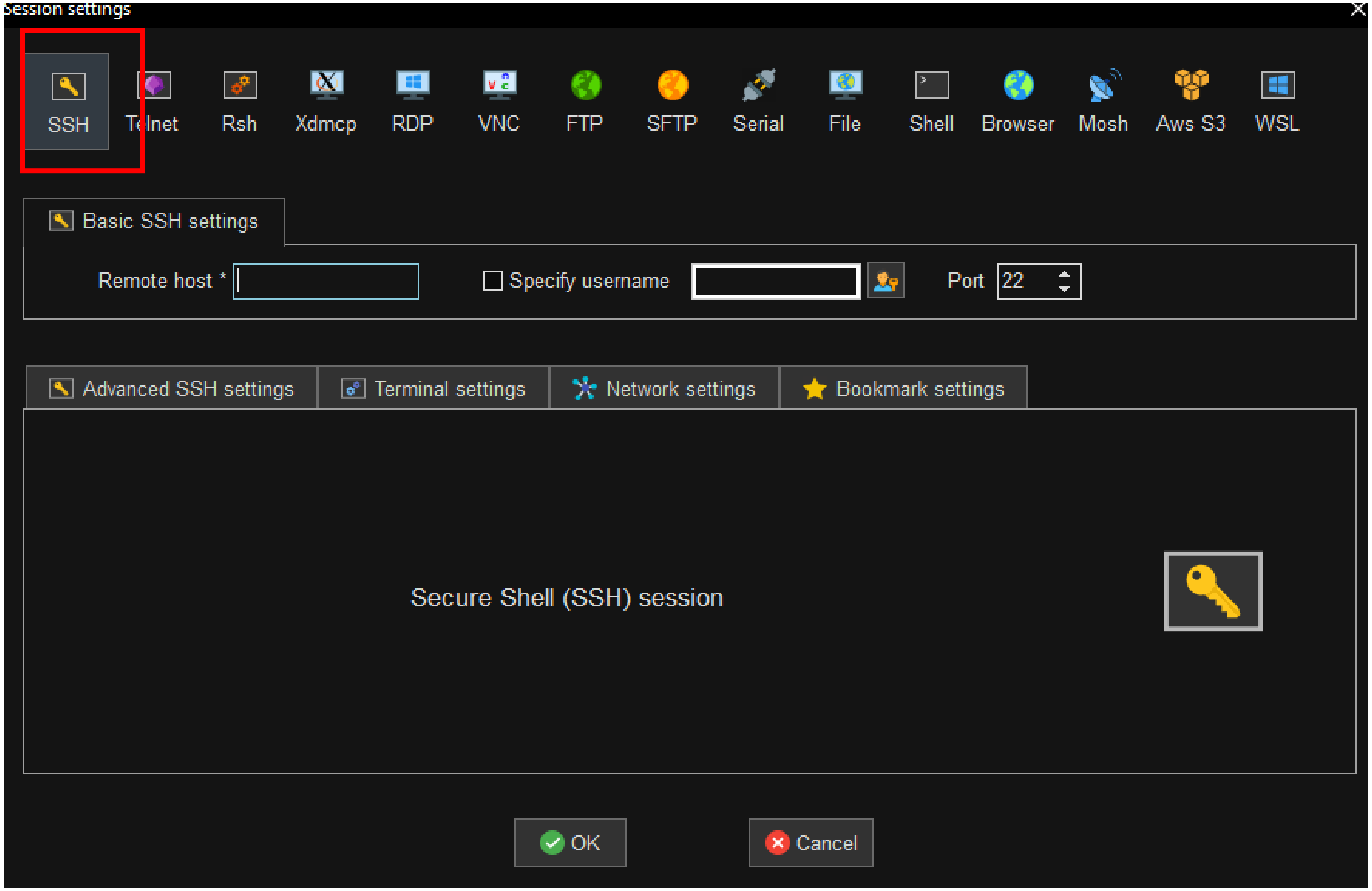
- Click Session



Access hoffman2 (Windows user)

Another option

- Click Session
- Click SSH



Remote host *

□ Specify username

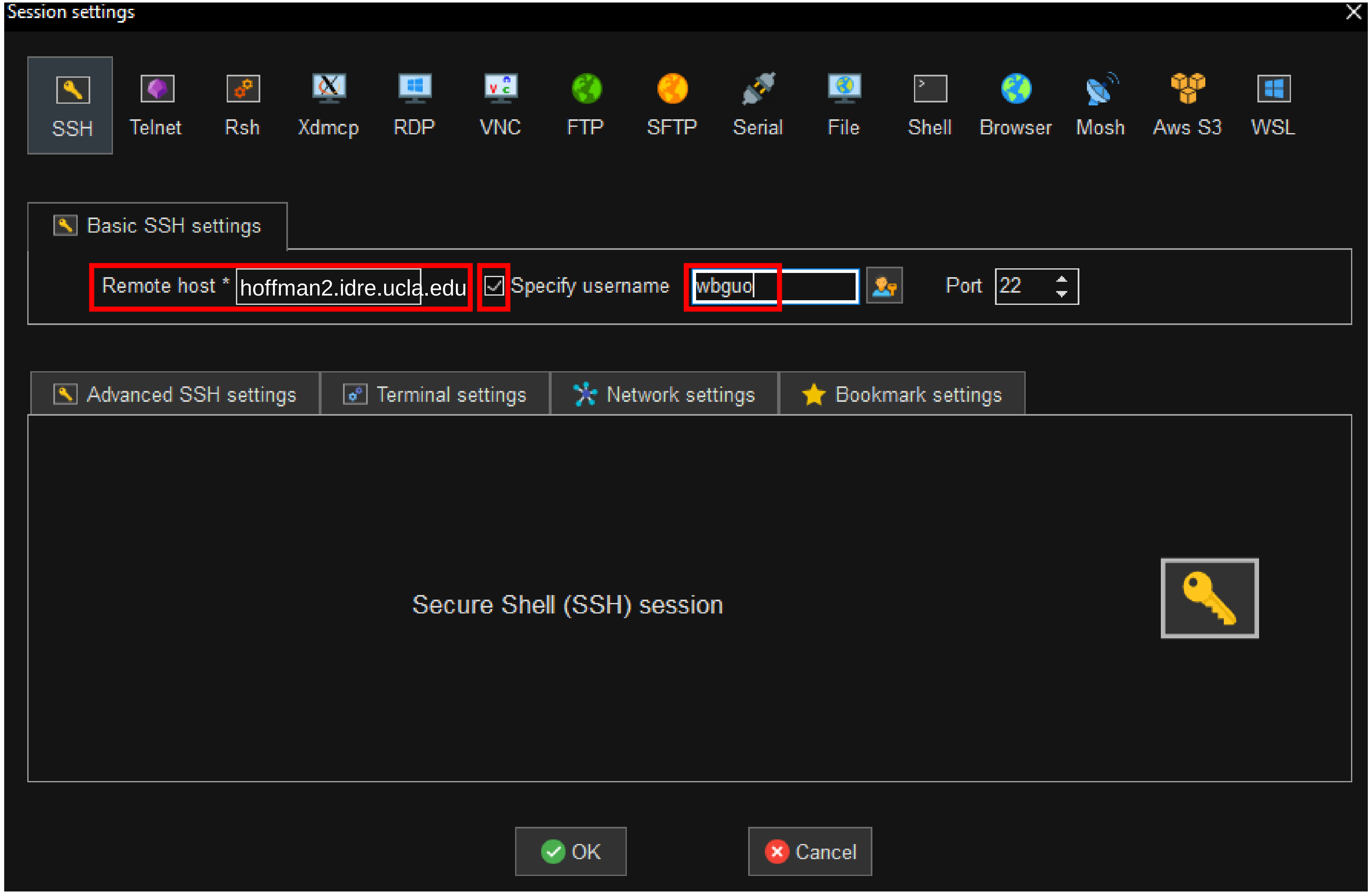
Port 22

Secure Shell (SSH) session

Access hoffman2 (Windows user)

Another option

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



Session settings

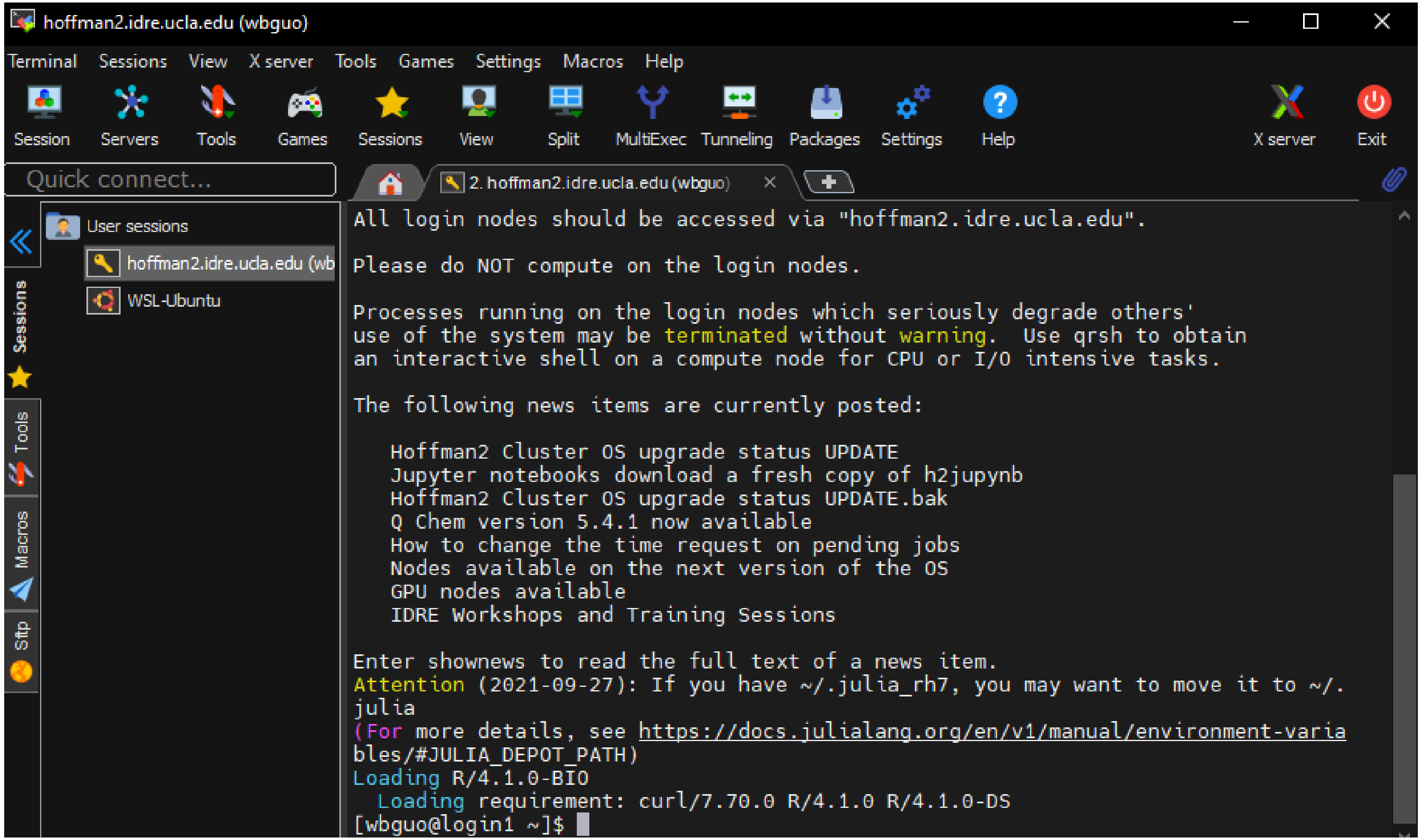
Basic SSH settings

Remote host * hoffman2.idre.ucla.edu ☑ Specify username wbguo Port 22

Access hoffman2 (Windows user)

Another option

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



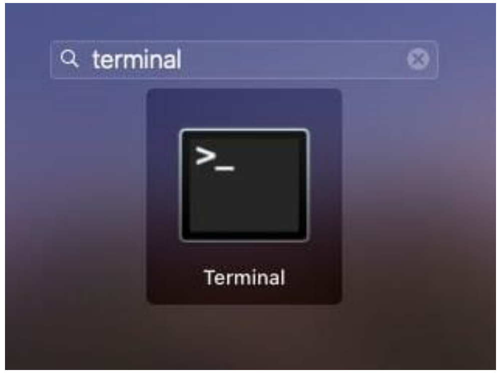
```
All login nodes should be accessed via "hoffman2.idre.ucla.edu".

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-varia
bles/#JULIA_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 ~]$
```

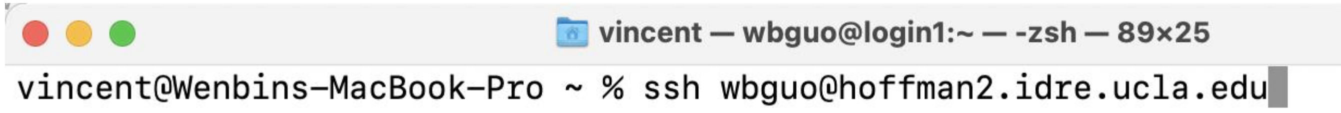
Access hoffman2 (macOS or Linux user)

Type in the following commands to your computer terminal:

- ➢ `ssh username@hoffman2.idre.ucla.edu`

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





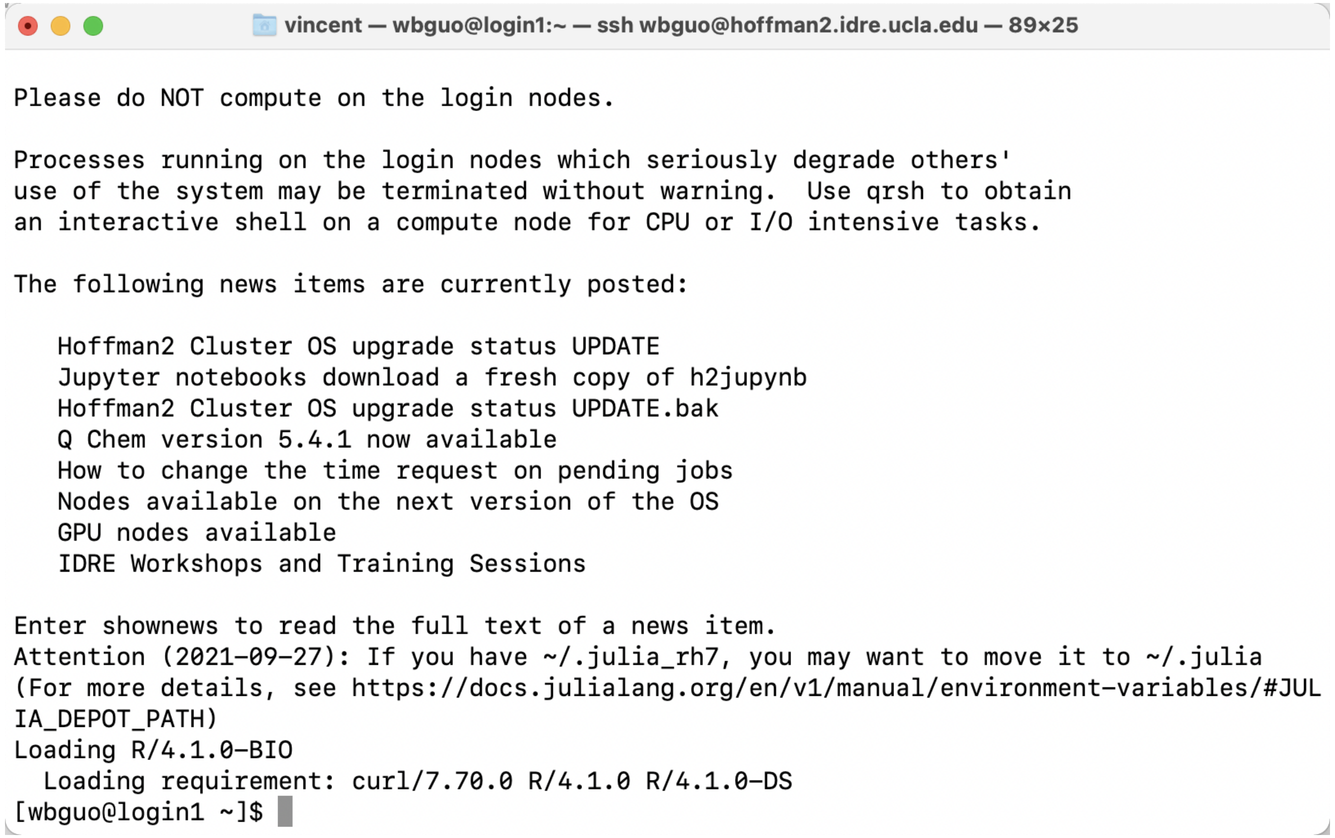
```
vincent@Wenbins-MacBook-Pro ~ % ssh wbguo@hoffman2.idre.ucla.edu
```

Access hoffman2 (macOS or Linux user)

Type in the following commands to your computer terminal:

- `ssh username@hoffman2.idre.ucla.edu`

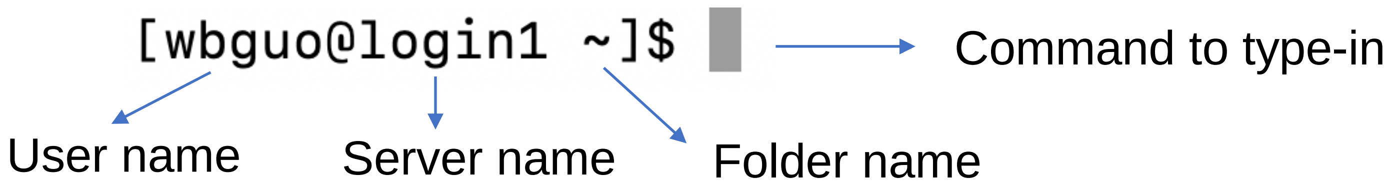
  (*replace* `username` *with your hoffman2 username, e.g. wbguo*)



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

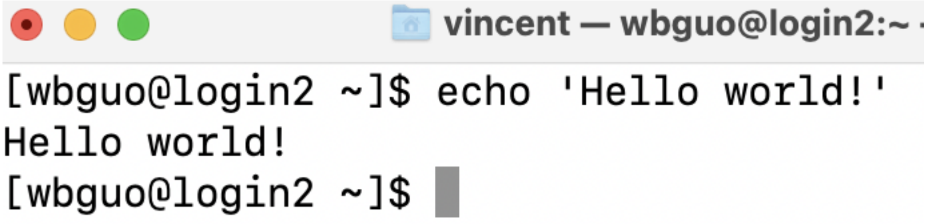
- It waits for your command and will execute it after you press **enter** key

- 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!'`

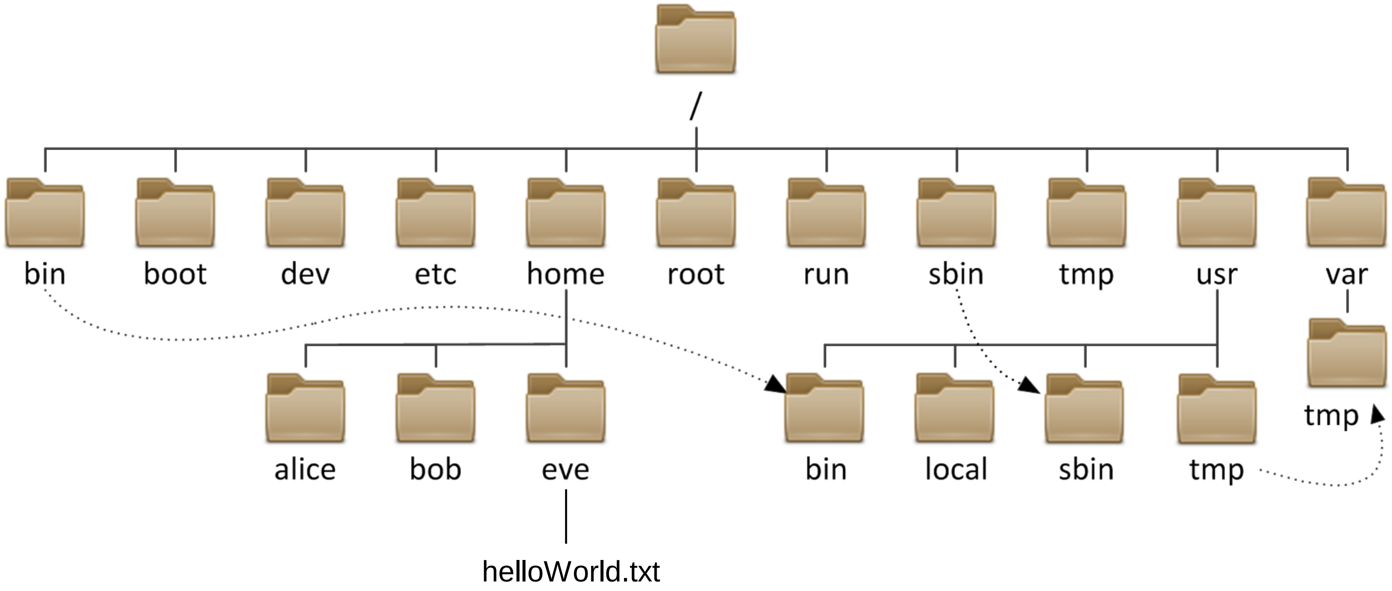


```
[wbguo@login2 ~]$ echo 'Hello world!'
Hello world!
[wbguo@login2 ~]$ 
```

  - `git clone https://github.com/wbvguo/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**

https://www.nationalgeographic.com/travel/national-parks/article/joshua-tree-national-park

Path to a file/directory

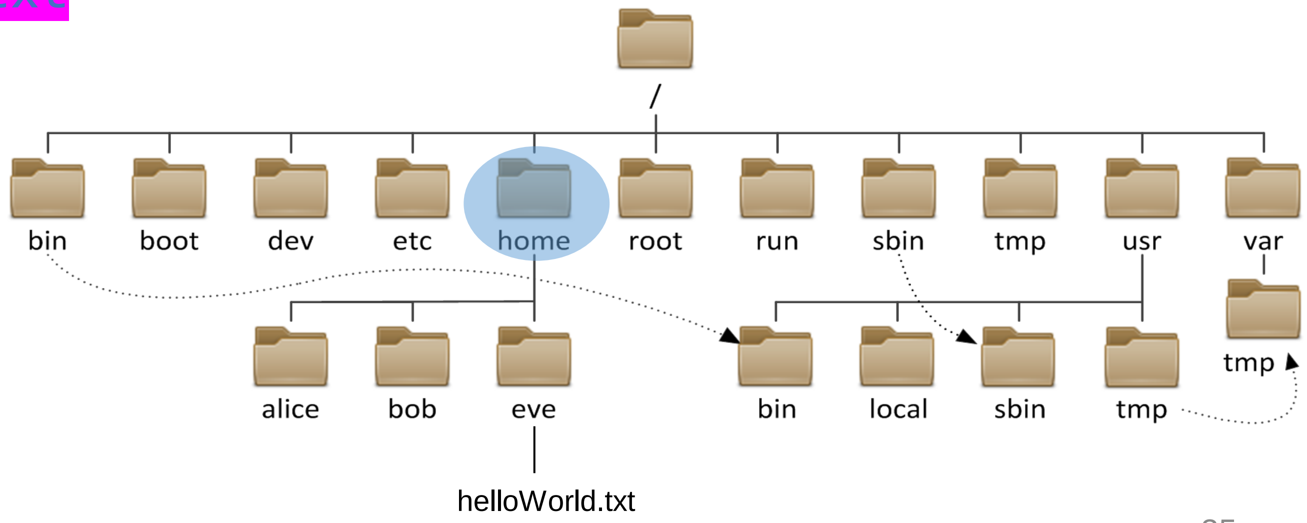
You can refer to a file/directory using absolute path or relative path

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

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

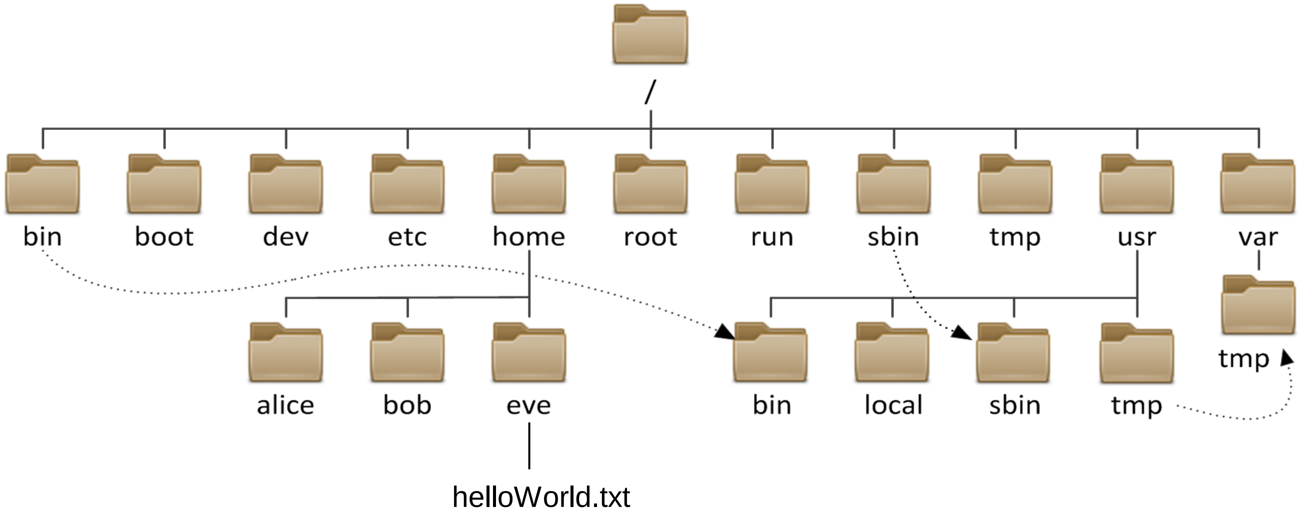
- Relative path: relative to your current location

  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

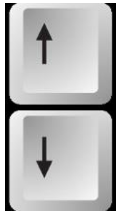
```
[wbguo@login1 qcbio-intro2Unix]$ ls -lSh
total 12K
drwxr-xr-x. 2 wbguo matteop 4.0K Jan  3 22:19 day1
drwxr-xr-x. 2 wbguo matteop 4.0K Jan  3 22:19 day2
drwxr-xr-x. 3 wbguo matteop 4.0K Jan  3 22:19 day3
-rw-r--r--. 1 wbguo matteop   19 Jan  3 22:19 README.md
```

- `-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

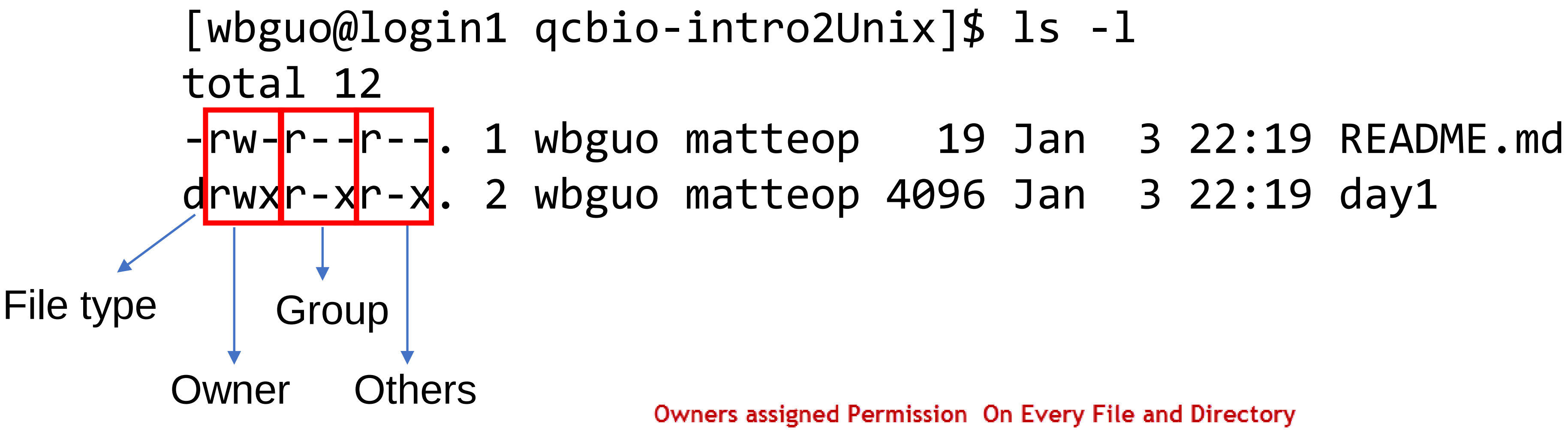
- Use `history` to see your command history

- Use `clear` to clean screen

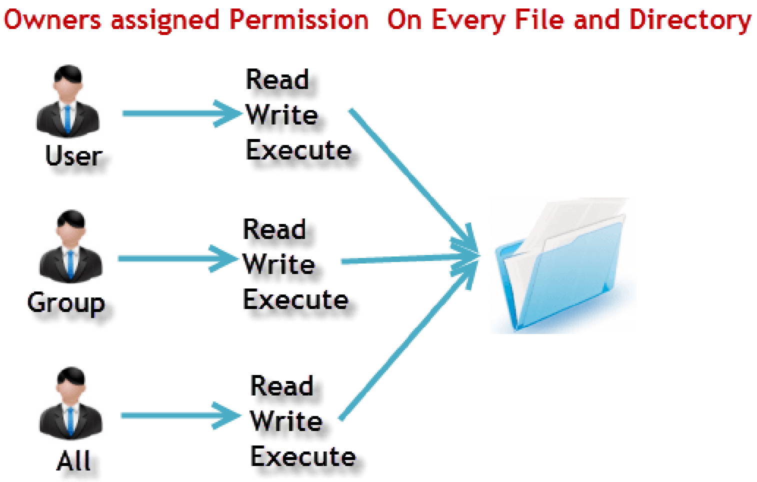
Practice

- 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



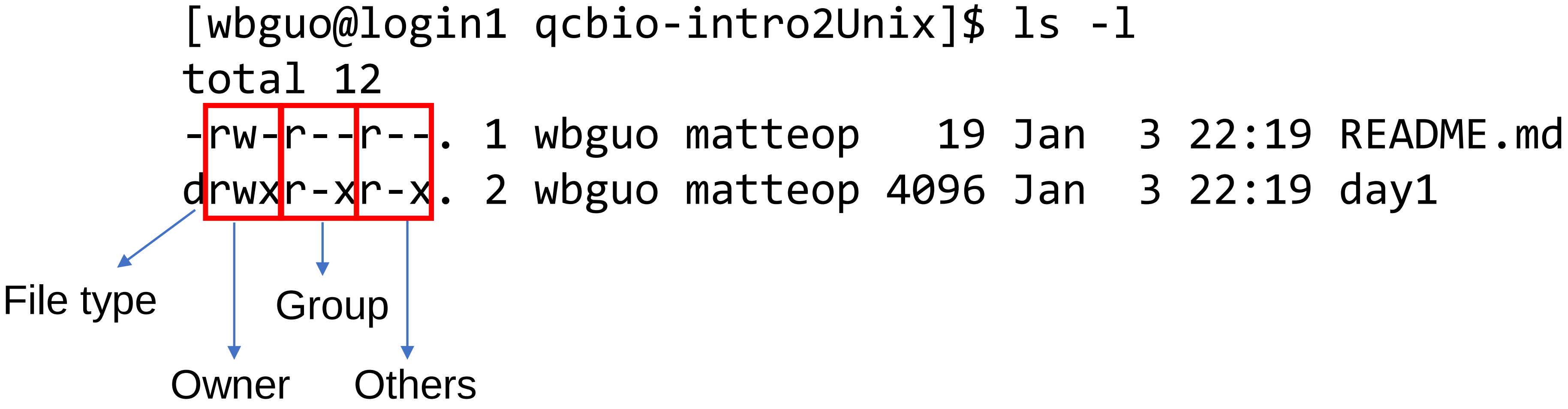
```
[wbguo@login1 qcbio-intro2Unix]$ ls -l
total 12
-rw-r--r--. 1 wbguo matteop   19 Jan  3 22:19 README.md
drwxr-xr-x. 2 wbguo matteop 4096 Jan  3 22:19 day1
```



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

https://www.guru99.com/file-permissions.html

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

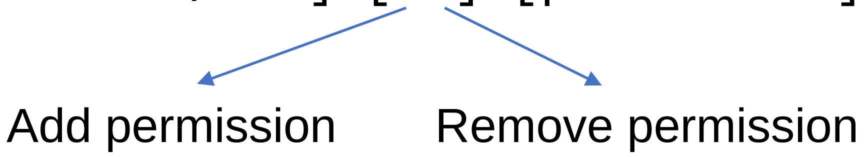
Owner

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>`

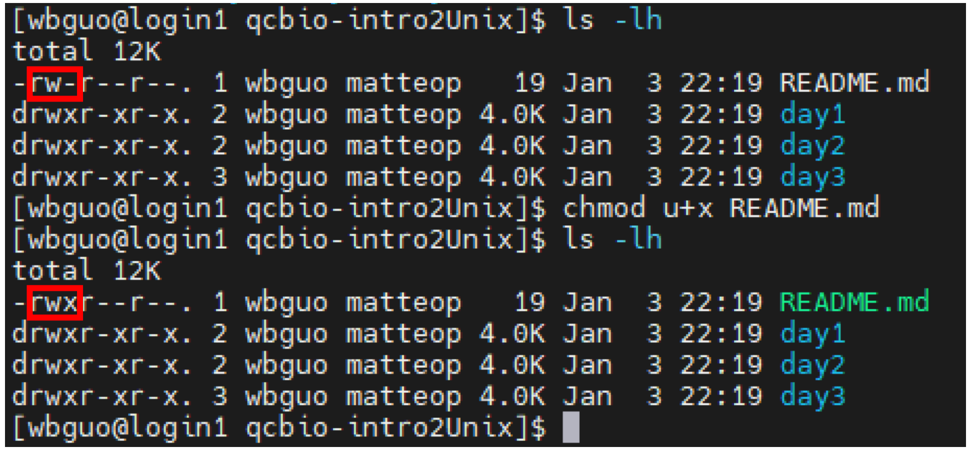


Add permission

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

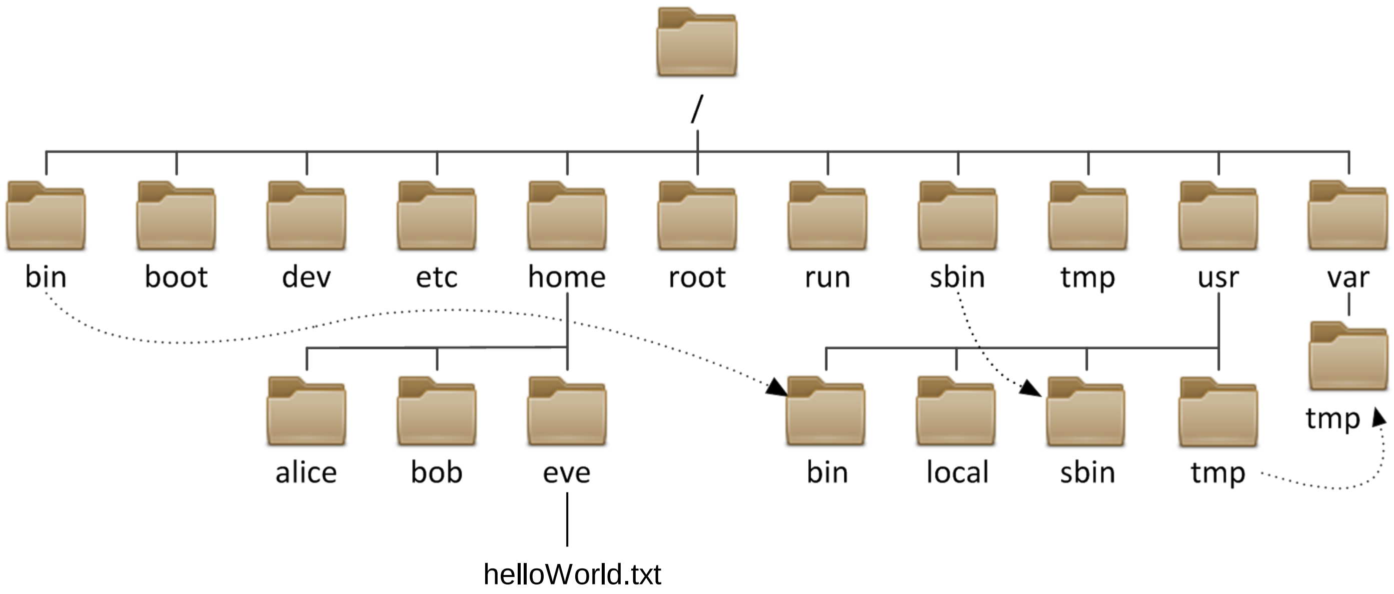
- Give README.md execution permission

- 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



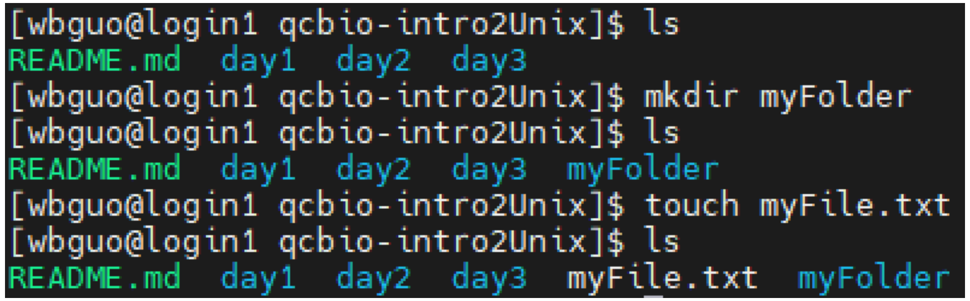
General syntax:

```
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>

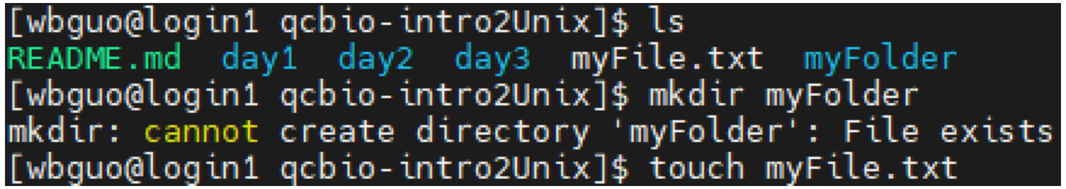


```
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md  day1  day2  day3
[wbguo@login1 qcbio-intro2Unix]$ mkdir myFolder
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md  day1  day2  day3  myFolder
[wbguo@login1 qcbio-intro2Unix]$ touch myFile.txt
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md  day1  day2  day3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

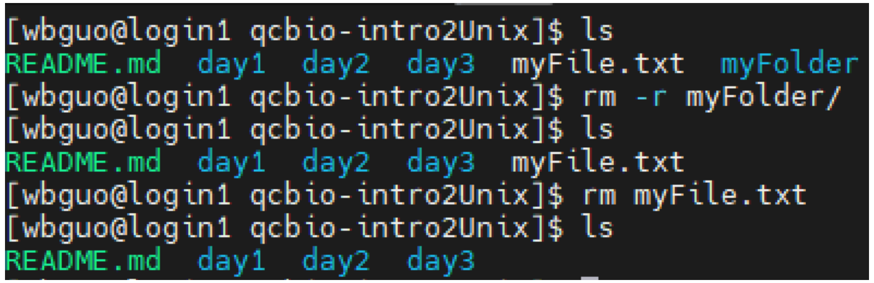
- Flag options:
  - -r: recursively

    Remove directory and their contents
  - -i: interactively

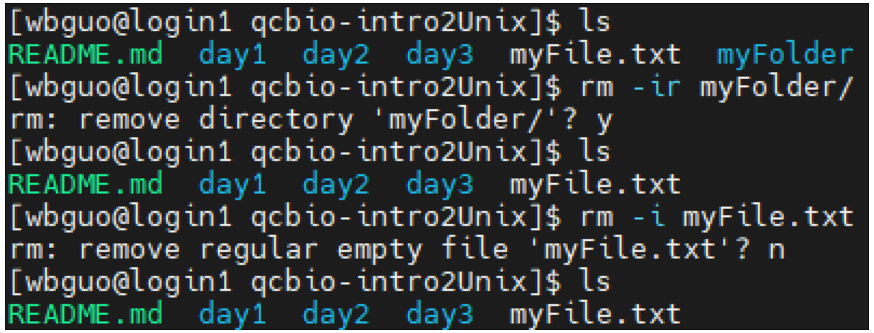
    Ask to confirm before every removal
  - -v: verbose

    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

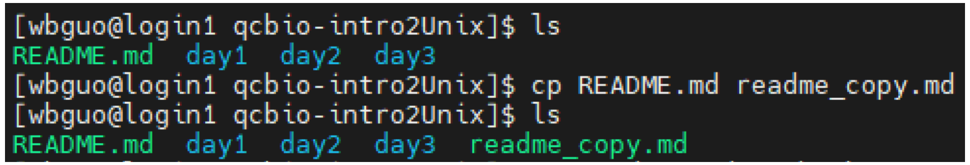
    Ask to confirm before overwrite
  - -v: verbose

    Explain what is being done

Copy (cp)

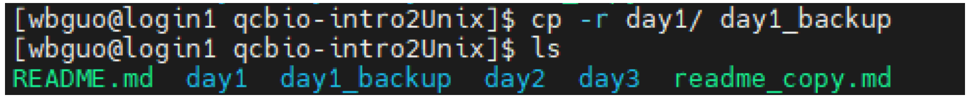
- `cp`: copy
- Syntax: `cp -<flag> <source> <destination>`

- Copy a file



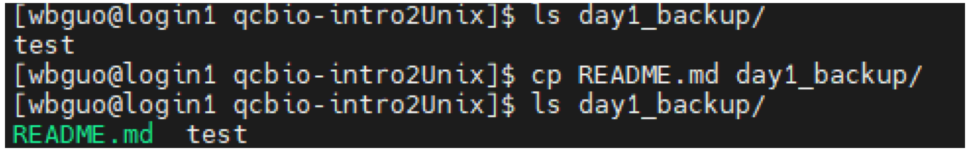
```
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md  day1  day2  day3
[wbguo@login1 qcbio-intro2Unix]$ cp README.md readme_copy.md
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md  day1  day2  day3  readme_copy.md
```

- 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

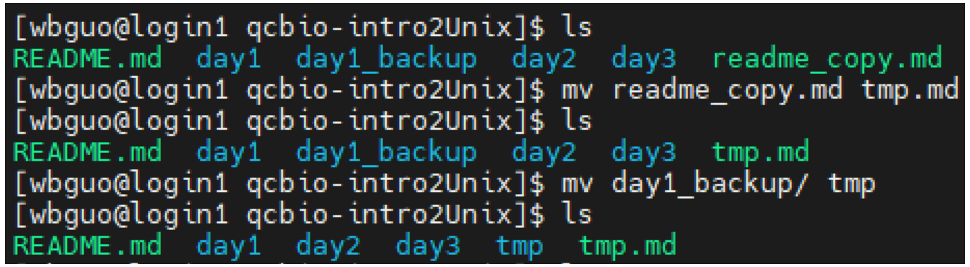


```
[wbguo@login1 qcbio-intro2Unix]$ ls day1_backup/
test
[wbguo@login1 qcbio-intro2Unix]$ cp README.md day1_backup/
[wbguo@login1 qcbio-intro2Unix]$ ls day1_backup/
README.md  test
```

Move (mv)

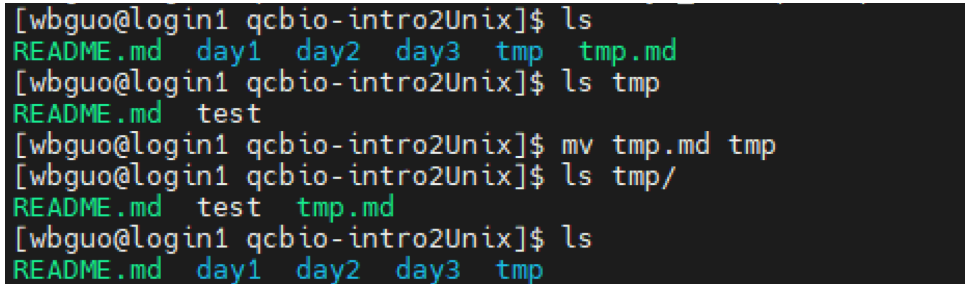
- mv: move
- Syntax: `mv <source> <destination>`

- 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

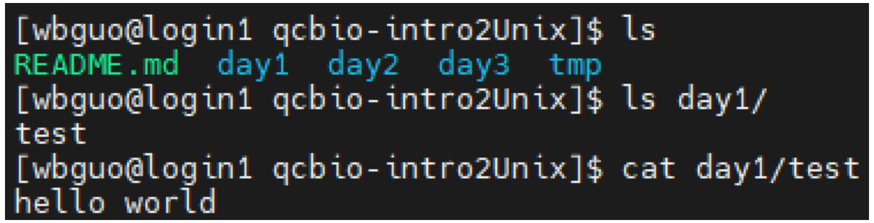
- Copy *day1* to *day1_backup*

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

- Rename *readme_copy.md* as *tmp.md*

View (cat)

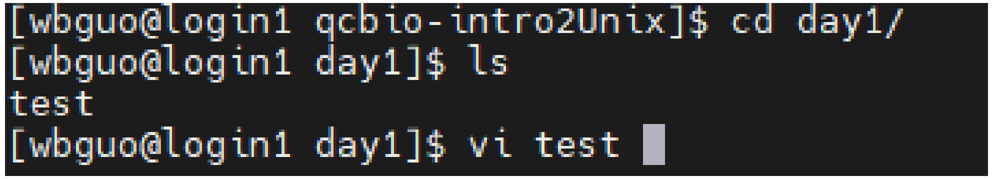
- `cat`: concatenate files and print on standard output
- Syntax: `cat <file name>`



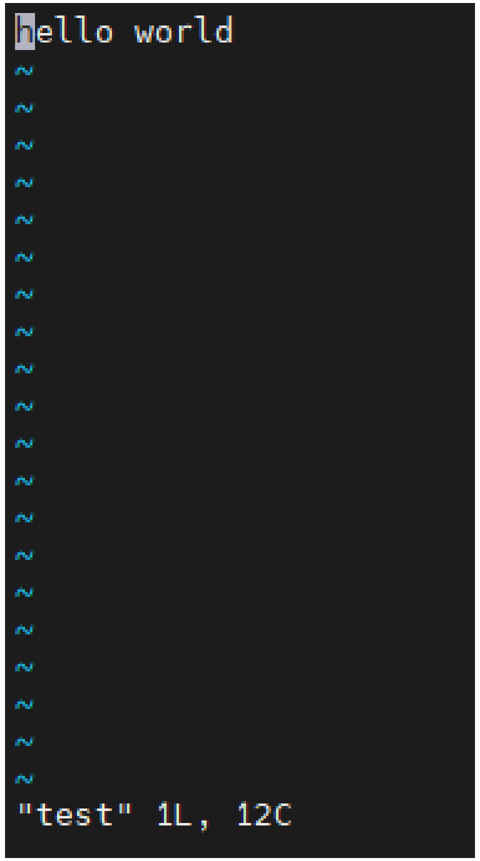
```
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md  day1  day2  day3  tmp
[wbguo@login1 qcbio-intro2Unix]$ ls day1/
test
[wbguo@login1 qcbio-intro2Unix]$ cat day1/test
hello world
```

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
```

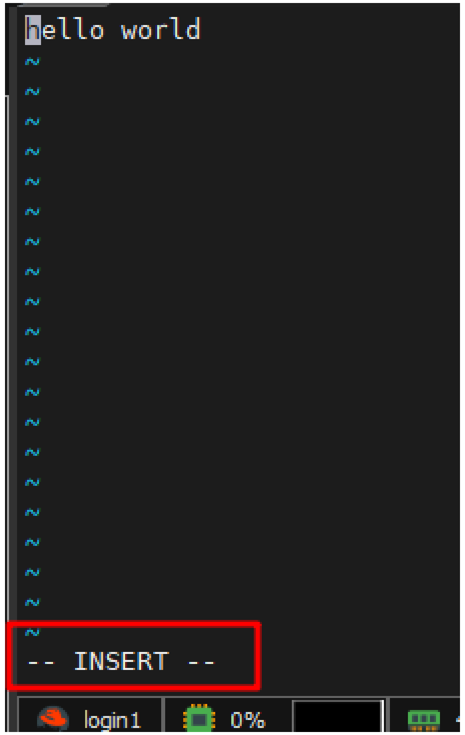


```
hello world
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"test" 1L, 12C
```

Modify (vi)

Once inside the file

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



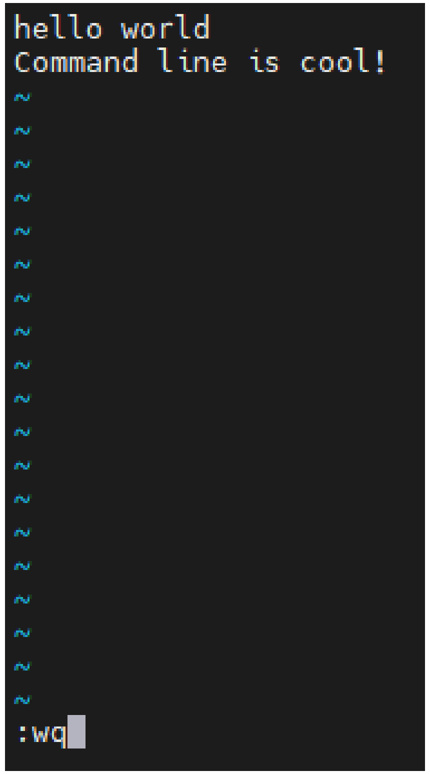
```
hello world
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

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)

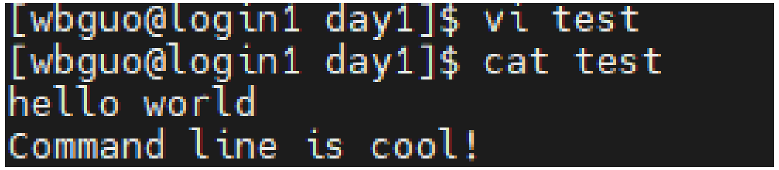


```
hello world
Command line is cool!
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

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

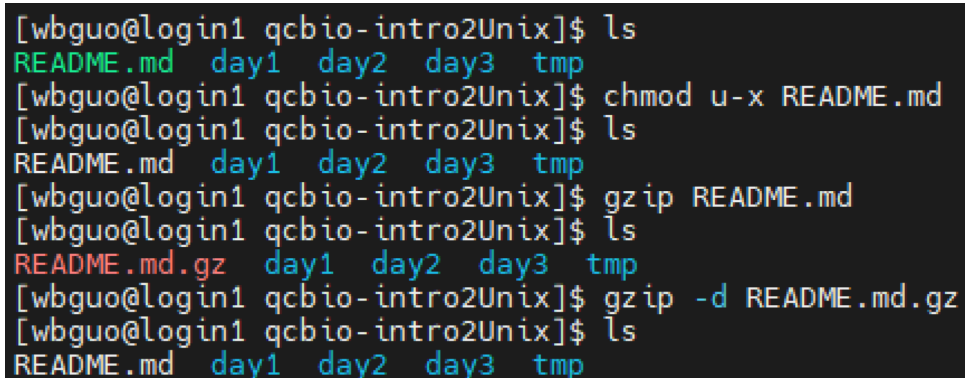
- Create a new file using vi, with contents “Less is more”

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

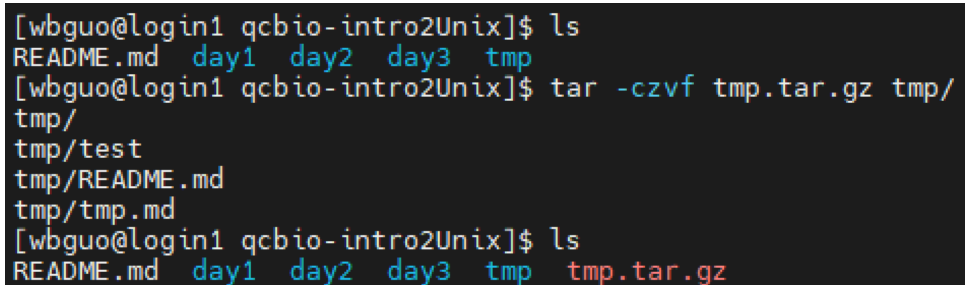
- `tar`: tape archive (create a compresed archive)
- Syntax:
  - Compress: `tar –czvf <file name>.tar.gz <file name>`
  - Decompress: `tar –xzvf <file name>.tar.gz`

- 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

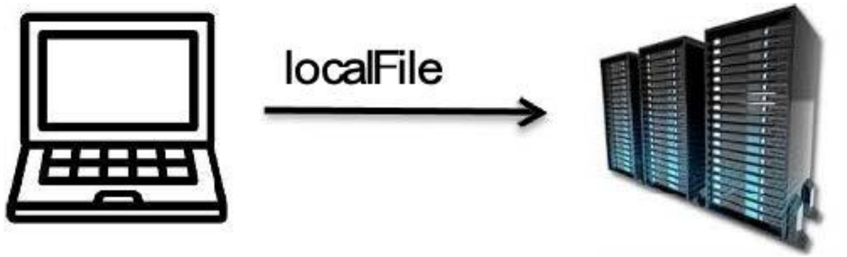
- `tar`: tape archive (create a compressed archive)
- Syntax:
  - Compress: `tar -czvf <file name>.tar.gz <file name>`
  - Decompress: `tar -xzvf <file name>.tar.gz`



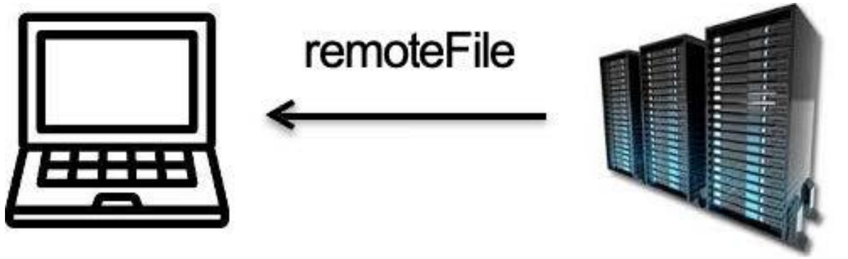
```
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md  day1  day2  day3  tmp
[wbguo@login1 qcbio-intro2Unix]$ tar -czvf tmp.tar.gz tmp/
tmp/
tmp/test
tmp/README.md
tmp/tmp.md
[wbguo@login1 qcbio-intro2Unix]$ ls
README.md  day1  day2  day3  tmp  tmp.tar.gz
```

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> .`

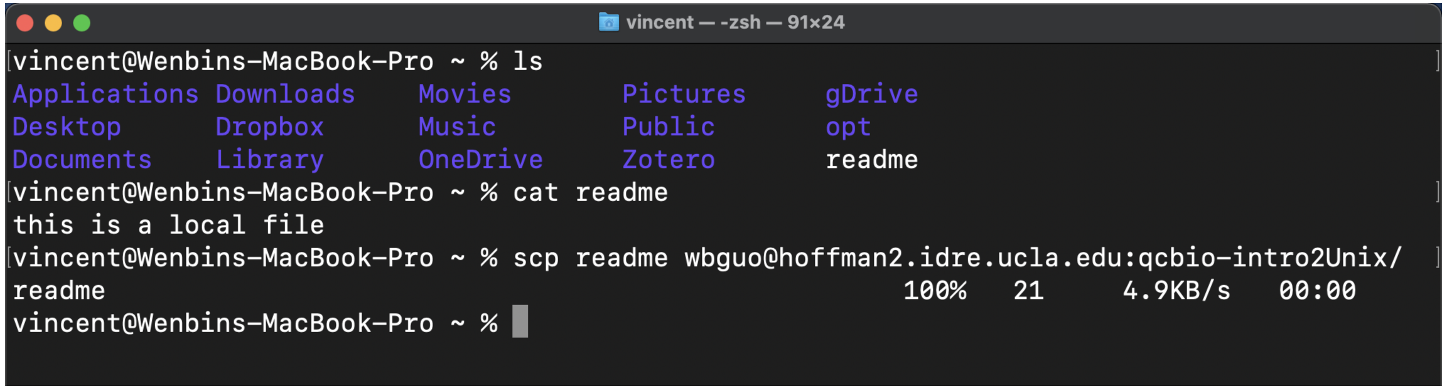


If you have a very large dataset to transfer, check hoffman2 data transfer option

Download & upload (`scp`)

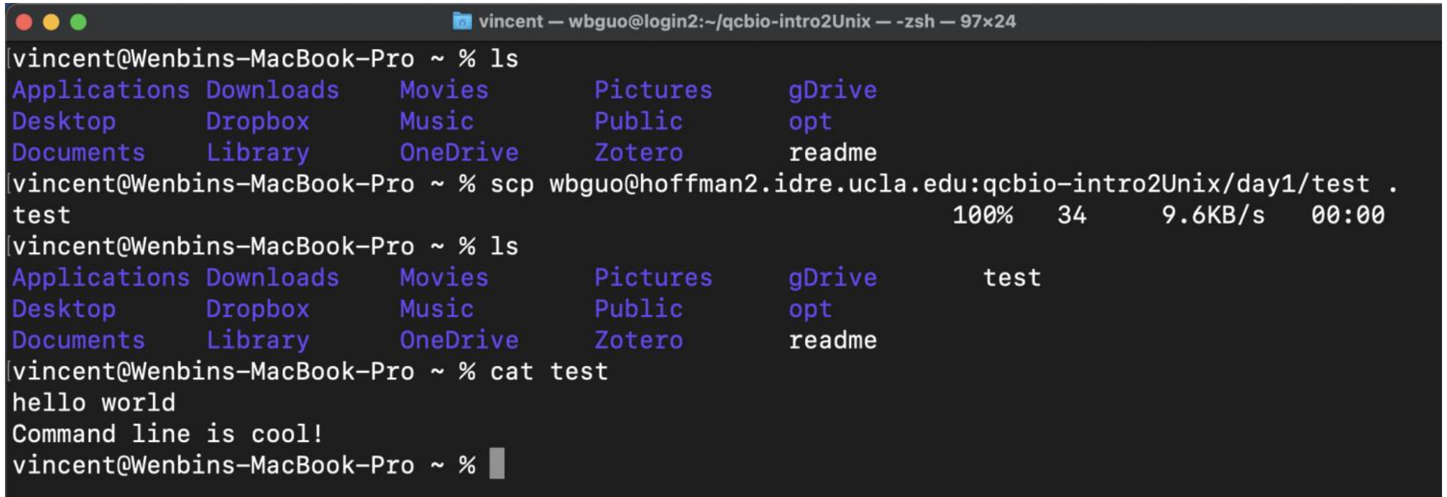
- scp: openssh secure file copy

- 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

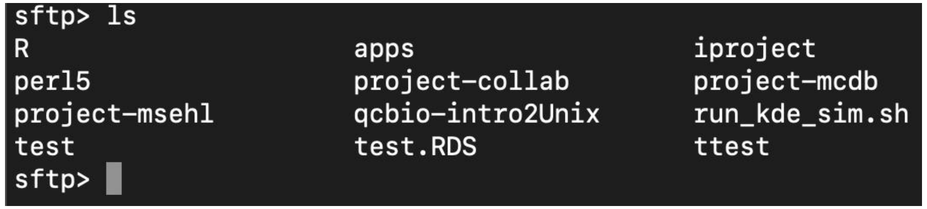


```
vincent@Wenbins-MacBook-Pro ~ % ls
Applications Downloads    Movies       Pictures     gDrive
Desktop      Dropbox      Music        Public       opt
Documents    Library      OneDrive     Zotero       readme
vincent@Wenbins-MacBook-Pro ~ % scp wbguo@hoffman2.idre.ucla.edu:qcbio-intro2Unix/day1/test .
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`)

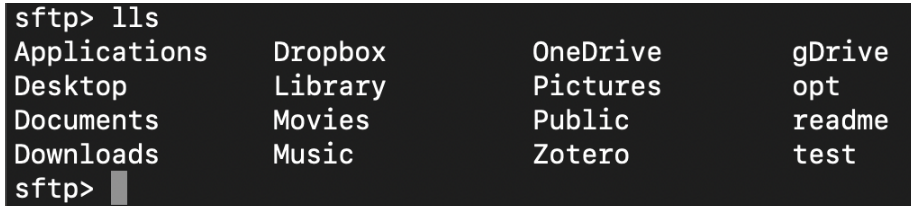
- sftp: (ssh file transfer protocol)
- Syntax (from local computer terminal):
  - `sftp username@hoffman2.idre.ucla.edu`

- List remote files: `ls`



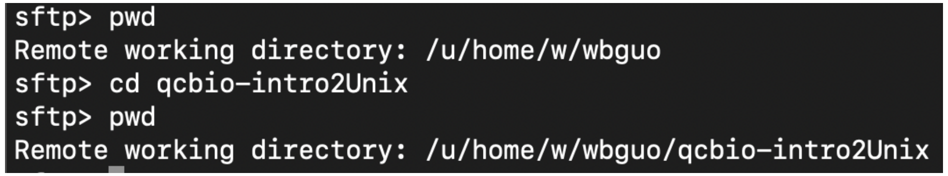
```
sftp> ls
R                     apps                  iproject
perl5                 project-collab        project-mcdb
project-msehl         qcbio-intro2Unix      run_kde_sim.sh
test                  test.RDS              ttest
sftp> 
```

- List local files: `lls`



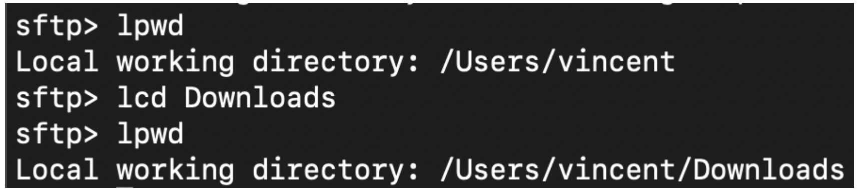
```
sftp> lls
Applications    Dropbox         OneDrive        gDrive
Desktop         Library         Pictures        opt
Documents       Movies          Public          readme
Downloads       Music           Zotero          test
sftp> 
```

- 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
```

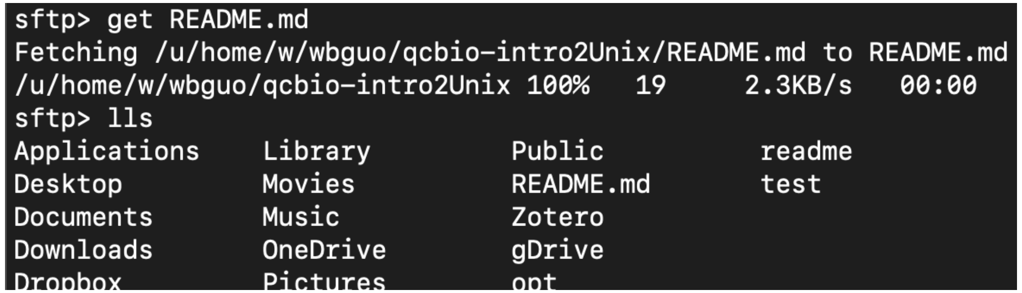
- Change local directory: `lcd`



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

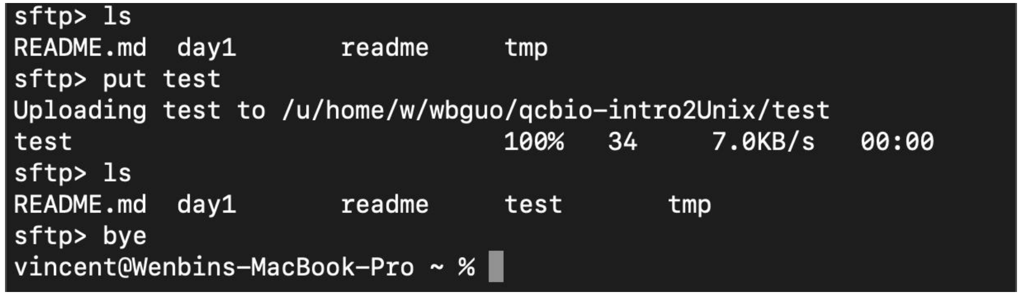
Download & upload (`sftp`)

- 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>`
  - Terminate sftp connection: `bye`

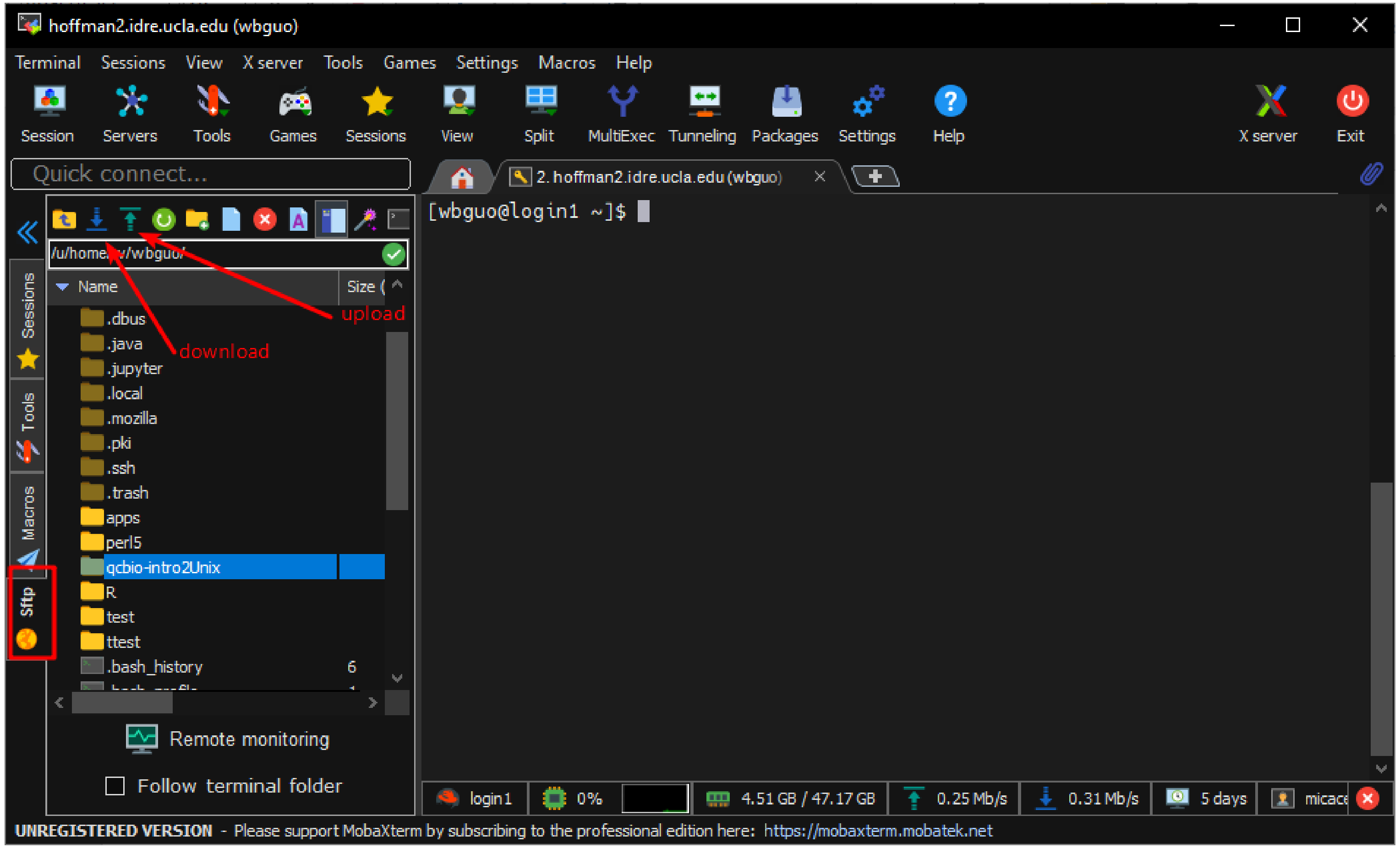


```
sftp> ls
README.md   day1        readme      tmp
sftp> put test
Uploading test to /u/home/w/wbguo/qcbio-intro2Unix/test
test                                100%   34     7.0KB/s   00:00
sftp> ls
README.md   day1        readme      test        tmp
sftp> bye
vincent@Wenbins-MacBook-Pro ~ % 
```

Download & upload (GUI tools option)

Windows user

- MobaXterm


hoffman2.idre.ucla.edu (wbguo)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit
Quick connect...
2. hoffman2.idre.ucla.edu (wbguo)
Sessions
Tools
Macros
Sftp
upload
download
Name
Size (
Remote monitoring
Follow terminal folder
login1
0%
4.51 GB / 47.17 GB
0.25 Mb/s
0.31 Mb/s
5 days
micace

/u/home/w/wbguo/

Name	Size (
.dbus	
.java	
.jupyter	
.local	
.mozilla	
.pki	
.ssh	
.trash	
apps	
perl5	
qcbio-intro2Unix	
R	
test	
ttest	
.bash_history	6

```
[wbguo@login1 ~]$ 
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

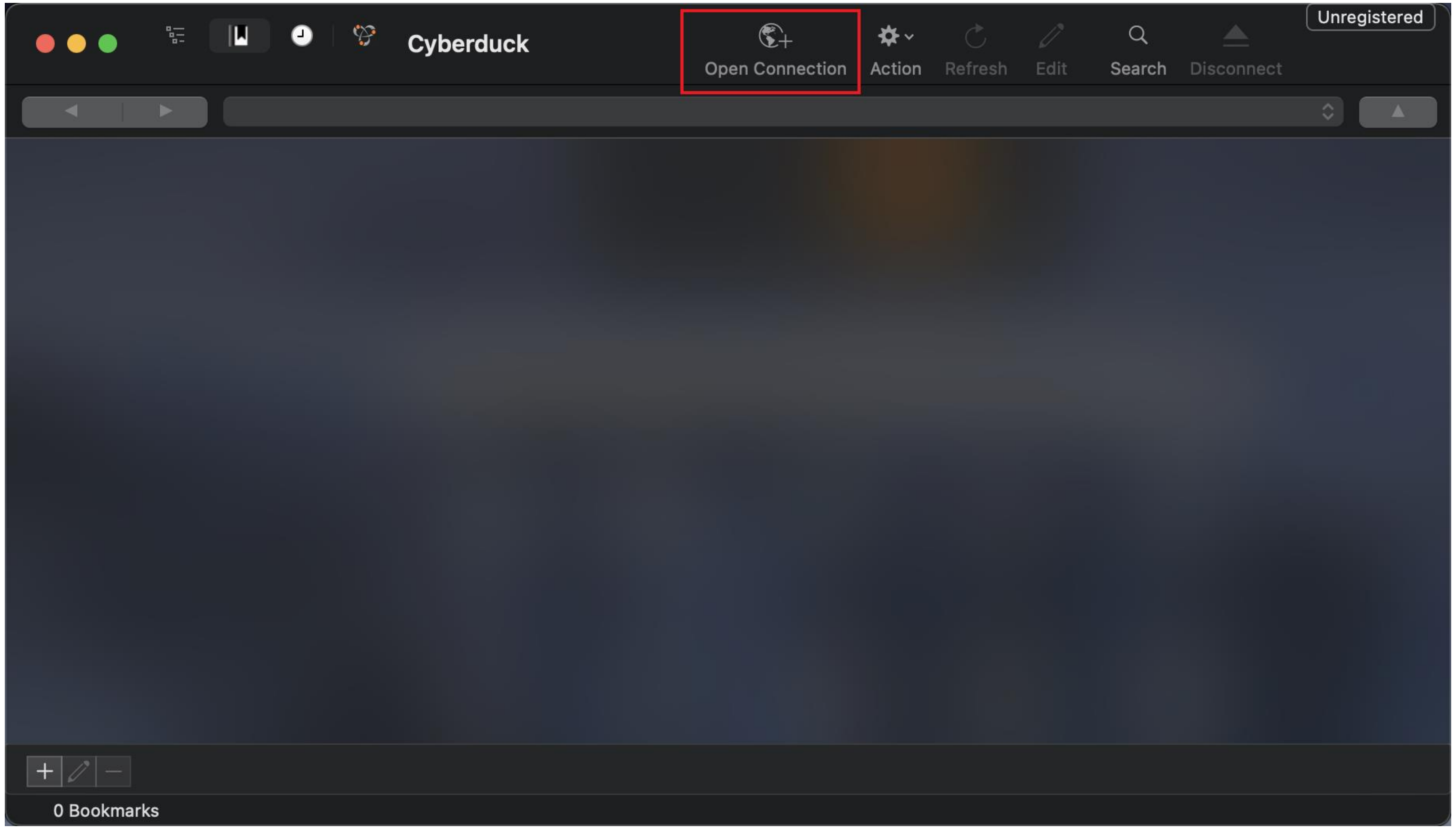
Other tools:

- WinSCP
- FileZilla

Download & upload (GUI tools option)

macOS user

- Cyberduck



Other tools:

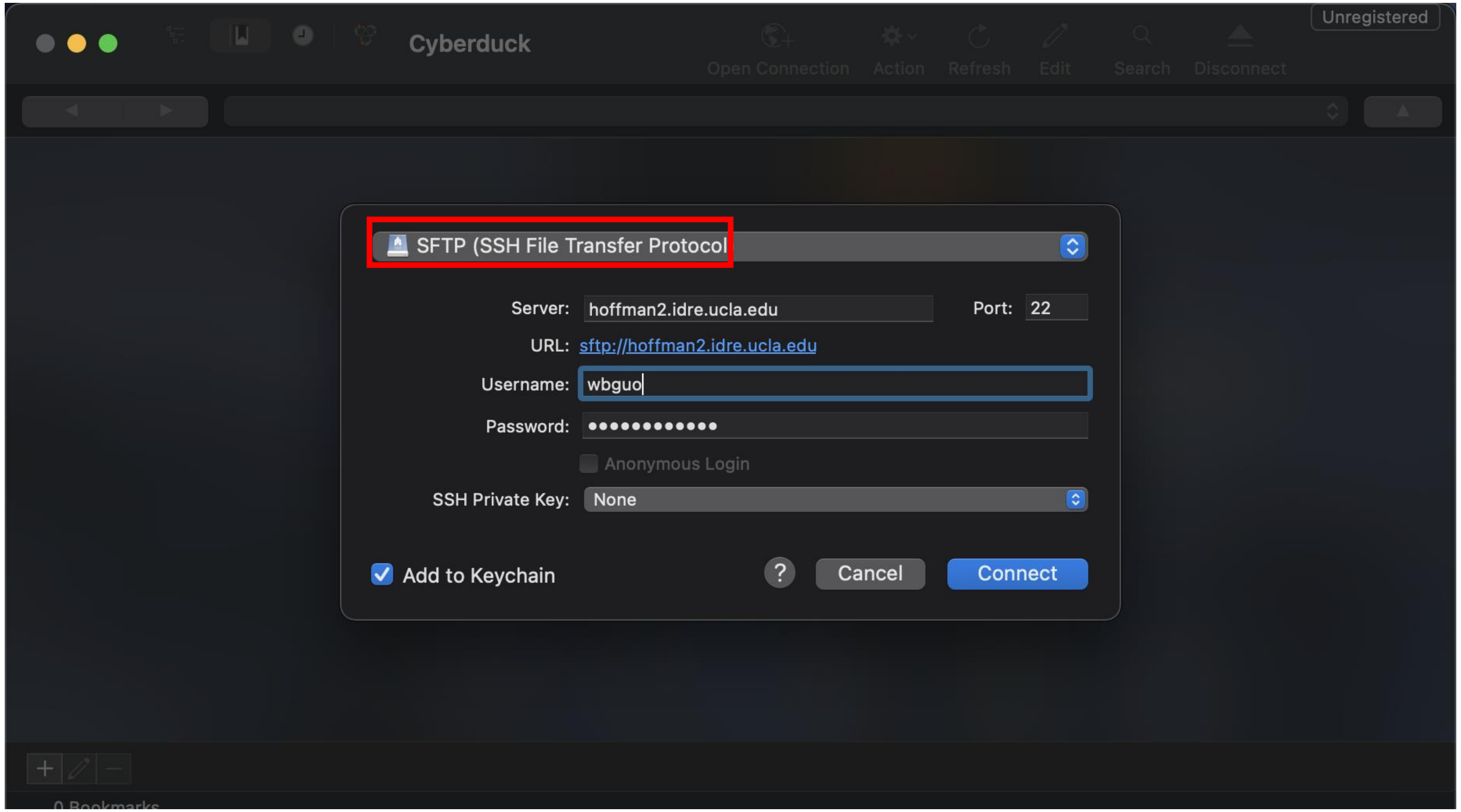
- FileZilla

Cyberduck is also available in windows

Download & upload (GUI tools option)

macOS user

- Cyberduck



SFTP (SSH File Transfer Protocol

Server: hoffman2.idre.ucla.edu

Port: 22

URL: sftp://hoffman2.idre.ucla.edu

Username: wbguo

Password: ●●●●●●●●●●●●

□ Anonymous Login

SSH Private Key: None

☑ Add to Keychain

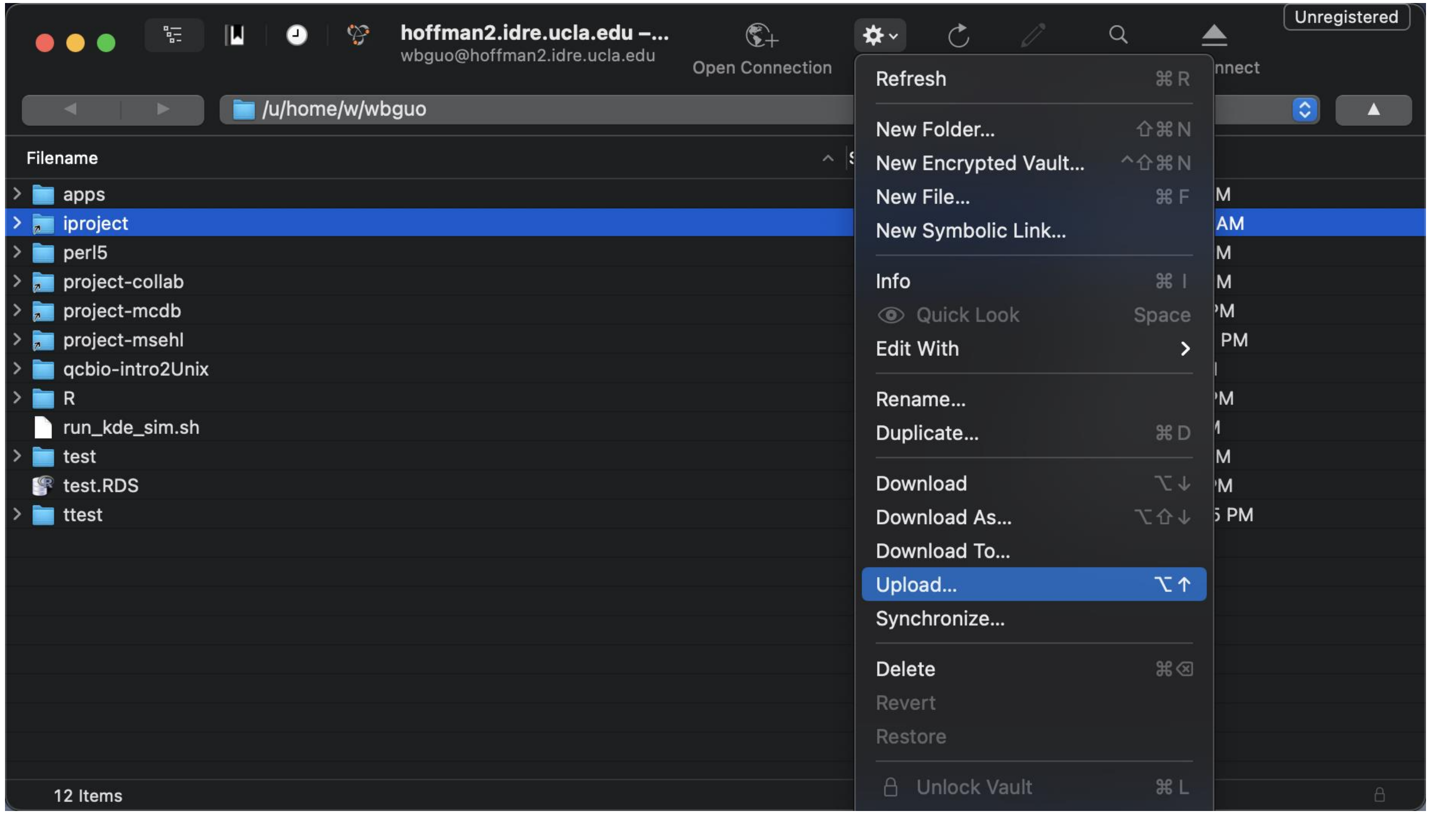
Other tools:

- FileZilla

Download & upload (GUI tools option)

macOS user

- Cyberduck



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

12 Items

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

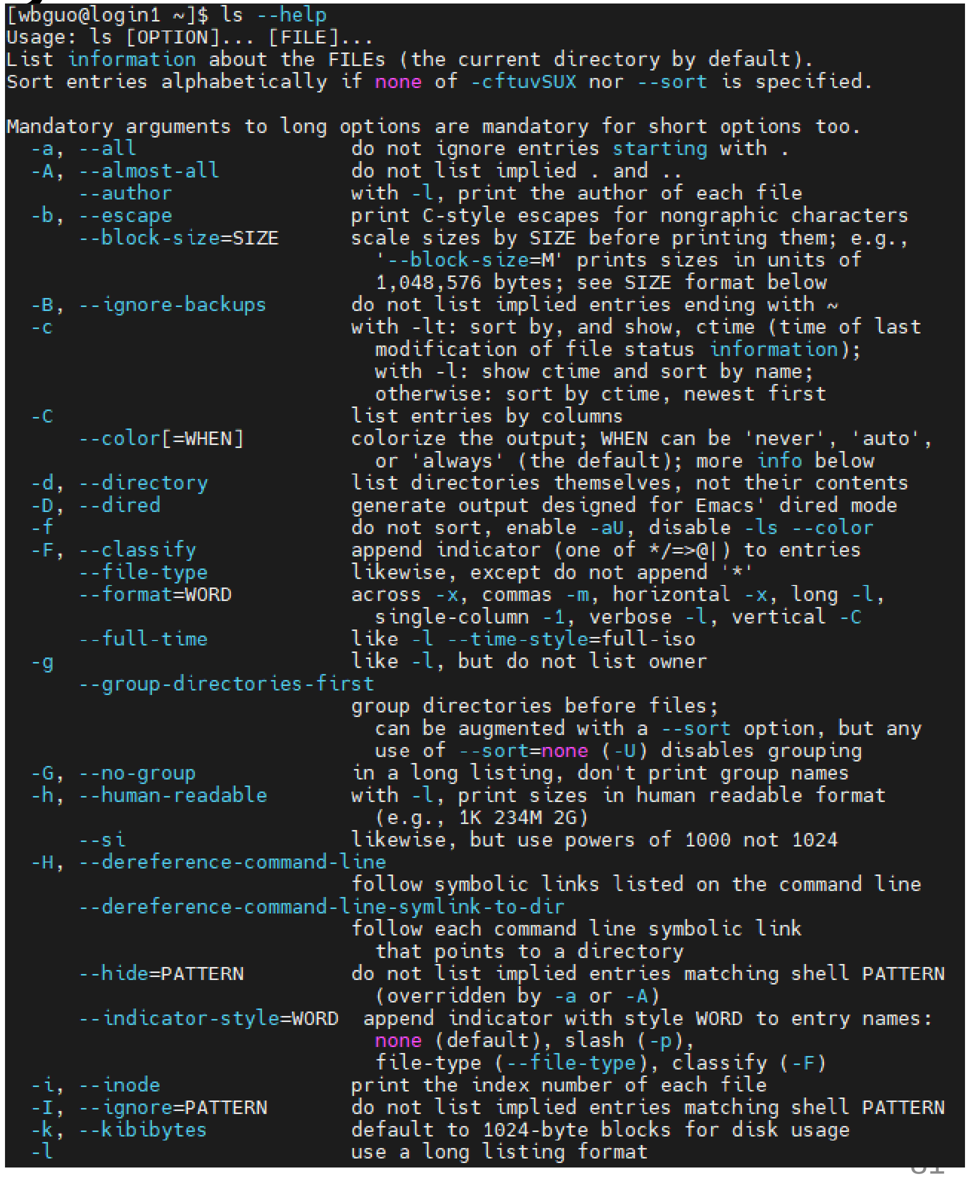
Close connection to hoffman2

- exit: close ssh connection

Cannot remember so many commands?

- No problem!

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



```
[wbguo@login1 ~]$ ls --help
Usage: ls [OPTION]... [FILE]...
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]         colorize the output; WHEN can be 'never', 'auto',
                               or 'always' (the default); more info below
  -d, --directory            list directories themselves, not their contents
  -D, --dired                generate output designed for Emacs' dired mode
  -f                         do not sort, enable -aU, disable -ls --color
  -F, --classify             append indicator (one of */=>@|) to entries
      --file-type            likewise, except do not append '*'
      --format=WORD          across -x, commas -m, horizontal -x, long -l,
                               single-column -1, verbose -l, vertical -C
      --full-time            like -l --time-style=full-iso
  -g                         like -l, but do not list owner
      --group-directories-first
                             group directories before files;
                               can be augmented with a --sort option, but any
                               use of --sort=none (-U) disables grouping
  -G, --no-group             in a long listing, don't print group names
  -h, --human-readable       with -l, print sizes in human readable format
                               (e.g., 1K 234M 2G)
      --si                   likewise, but use powers of 1000 not 1024
  -H, --dereference-command-line
                             follow symbolic links listed on the command line
      --dereference-command-line-symlink-to-dir
                             follow each command line symbolic link
                               that points to a directory
      --hide=PATTERN         do not list implied entries matching shell PATTERN
                               (overridden by -a or -A)
      --indicator-style=WORD  append indicator with style WORD to entry names:
                               none (default), slash (-p),
                               file-type (--file-type), classify (-F)
  -i, --inode                print the index number of each file
  -I, --ignore=PATTERN       do not list implied entries matching shell PATTERN
  -k, --kibibytes            default to 1024-byte blocks for disk usage
  -l                         use a long listing format
```

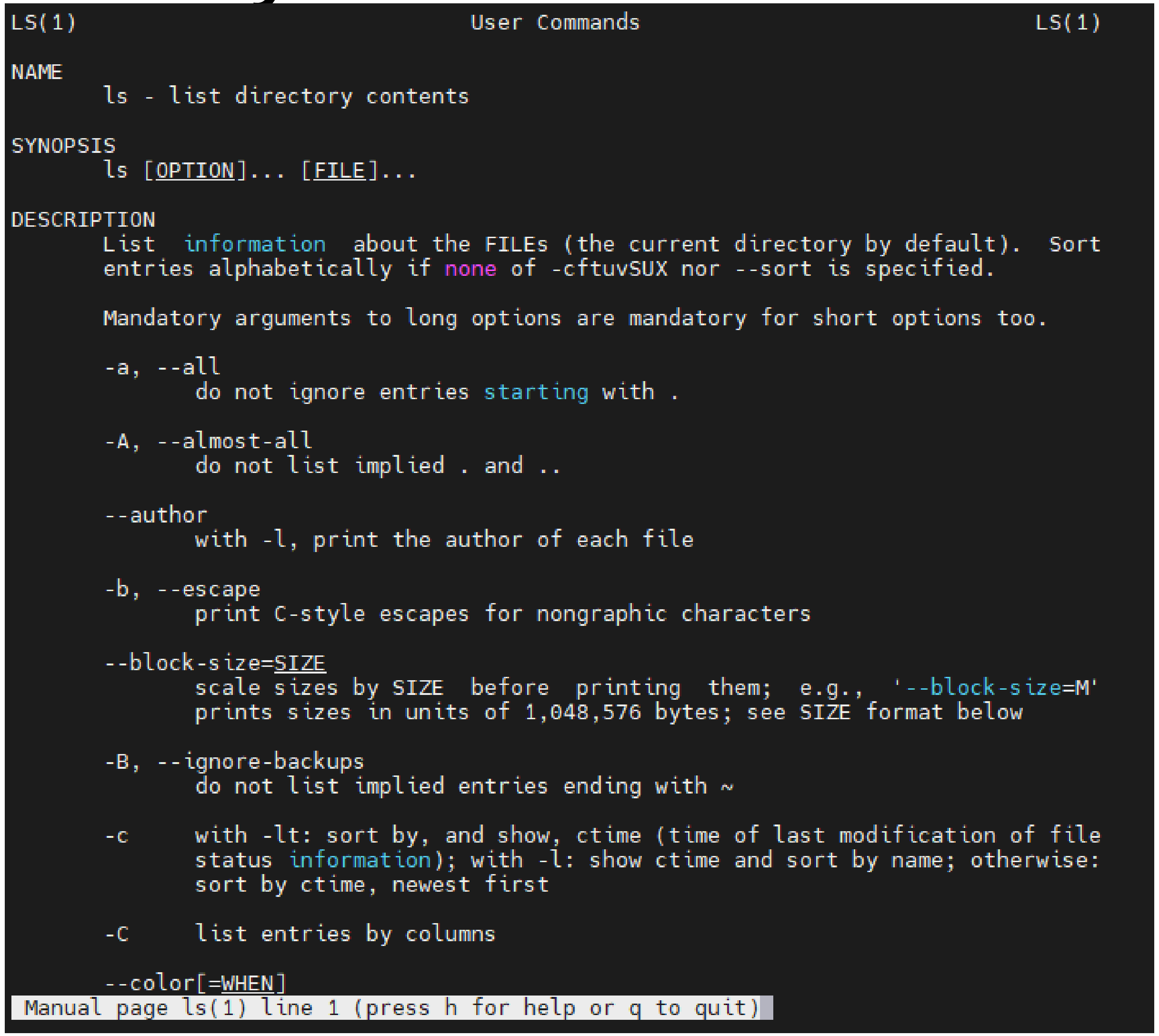
Cannot remember so many commands?

- 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

Q&A

Google doc