

# Welcome to CS35L!

**“Software Construction Laboratory”**

# Logistics

## ❖ Tameez Latib

- [tameezlatib@gmail.com](mailto:tameezlatib@gmail.com), please add “CS35L” to the subject line
- Office Hours: ?? Monday 4pm-6pm
- Feedback: <https://forms.gle/6kcJ2aJtzAzFMhHQ7> (anonymous google form)
- I took this class during undergrad

## ❖ CS 35L is unique

- TA lectures
- Weekly assignments <https://web.cs.ucla.edu/classes/fall20/cs35L/assign.html>
  - You can start early!

## ❖ CS 97

- Prototype course by professor Eggert
- Equivalent credit to CS 35L
- Email me (or Eggert) if you want to know more

# Syllabus, Grading, etc.

- ❖ Everything is here: <https://web.cs.ucla.edu/classes/fall20/cs35L/index.html>
- ❖ Each week has different content, mostly modular
- ❖ Each assignment is worth 5% of final grade
  - Can turn in late, but lose % points exponentially (1, 2, 4, ...)
- ❖ Final exam is 45%
  - Open everything, no collaboration
- ❖ New: 5% participation grade
  - Weekly quiz, unlimited attempts
- ❖ Final assignment is a presentation

# Other

## ❖ Problems with CS 35L

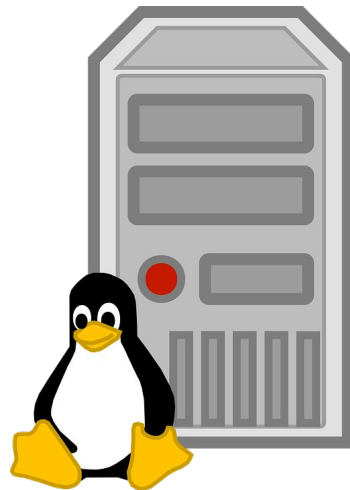
- Each TA teaches differently
  - Recorded lectures on CCLE
- Content may seem 'boring'
  - This class is extremely useful, but you may not notice it until later
- **It's a LOT of work**
  - Try to start early
  - Come to office hours / ask TAs for help / ask on piazza
  - Don't stress, use the lateness policy if needed
- Because of the workload, people cheat
  - Standard rules apply, study groups encouraged, but work individually
    - For more info, see <https://web.cs.ucla.edu/classes/fall20/cs35L/grading.html>

# Cheating

- ❖ You're here to learn
- ❖ Use the lateness policy!
- ❖ If you are really stuck, come to office hours!
- ❖ You may get caught.
  - Is it really worth it?
- ❖ You may develop bad habits
- ❖ Engineering ethics
- ❖ Part of this class is learning to learn

# Moving on... Linux

- ❖ Linux is a kernel, the core of an operating system (OS)
  - Allocates system resources to running programs
  - Connects hardware to software
- ❖ GNU software + Linux kernel → GNU/Linux OS
- ❖ Linux is free, open-source, and has many distributions: Ubuntu, Arch, Red Hat
- ❖ Why?
  - Stable, reliable, cheap, etc
- ❖ Why should you care?
  - You will be working with linux



# How to work with linux: CLI vs GUI

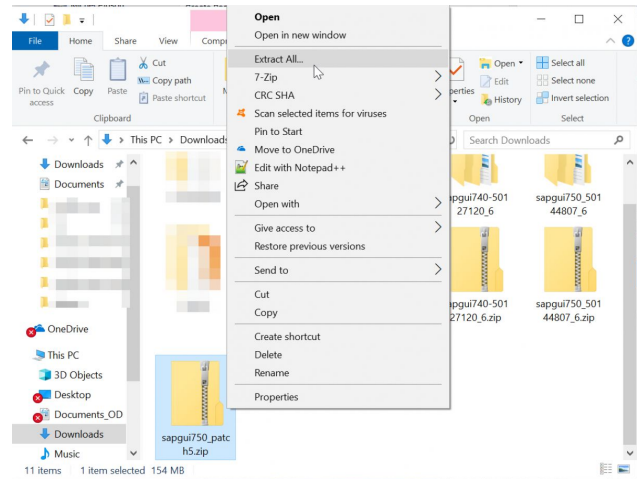
## ❖ Graphical User Interface (GUI)

- Applications, Graphics, etc.
  - Designed for user experience
- Easy to use but limited

## ❖ Command Line Interface (CLI)

- Shell program, scripting
- Learning curve
  - Greater flexibility
- What you will be working with

## ❖ Example: File explorer vs “ls / cd” commands



```
[root@desshi ~]#  
[root@desshi ~]# mkdir directorio0  
[root@desshi ~]# ls  
anaconda-ks.cfg  directorio0  install.log  rpmbuild  
Desktop          home         install.log.syslog  
[root@desshi ~]# cd directorio0/  
[root@desshi directorio0]# ls  
[root@desshi directorio0]# pwd  
/root/directorio0  
[root@desshi directorio0]#
```

# Seasnet, Inxsrv and how to access:

- ❖ This quarter, we require UCLA VPN
  - <https://www.it.ucla.edu/it-support-center/services/virtual-private-network-vpn-clients>
- ❖ Seasnet account
  - If you do not have a seasnet account, please sign up ASAP
  - <https://www.seasnet.ucla.edu/seasnet-accounts/>



# Seasnet, Inxsrv and how to access:

- ❖ Login to the VPN with your my.ucla username/password
  - May need duo as well
- ❖ MacOS/Linux
  - Use terminal and type 'ssh [SEASNETuser@Inxsrv10.seas.ucla.edu](https://seasnetuser@inxsrv10.seas.ucla.edu)'
  - Use your own personal SEASNET username in the command above
  - Enter SEASNET password when prompted
- ❖ Windows
  - Download a ssh client (PuTTY is popular).
  - When open, specify the Host Name as Inxsrv10.seas.ucla.edu, port 22, and connection type as SSH
  - Enter SEASNET username and password when prompted
- ❖ NOTE – in this class we will use Inxsrv06, 07, 09, and 10
- ❖ Demo

# Seasnet, Inxsrp and how to access:

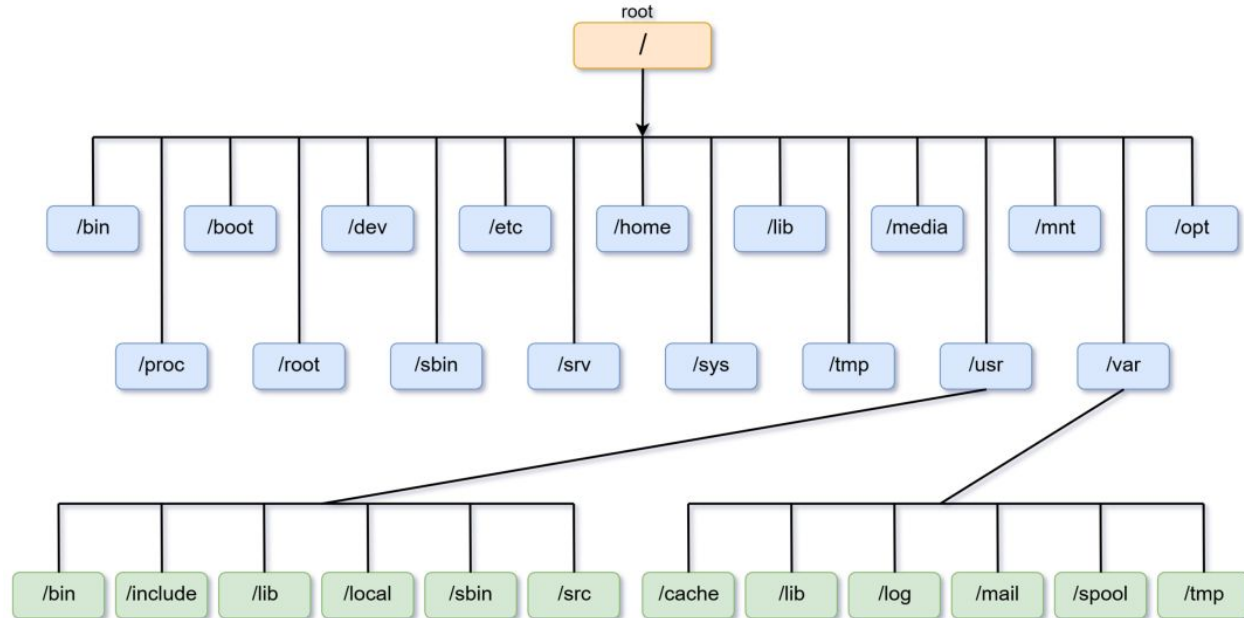
## ❖ Remote desktop

- <https://www.seasnet.ucla.edu/setting-up-remoteapps-and-remote-desktop/>
- More graphical
  - Can use apps like VS Code
  - Won't need to scp (copy from server to local, vice versa)
- You will still need to use CLI
- Can be very slow

# File or a process

- ❖ Everything is a file. If data is stored, it is a file.
  - This includes text documents, images, directories (folders), etc
    - Even for directories, need to store path, name, metadata
- ❖ A process is a program that is currently running
  - You can check task manager to see what's currently running on your computer
    - Zoom, chrome, etc

# Linux File Layout - Tree Structure



# File paths

## ❖ Absolute path:

- `/usr/local`

## ❖ Relative path, you're already in `/usr`

- `./local` = `/usr/local`
- `.` = current directory
- `../` = `/` (root)
- `..` = parent directory
- `~` = home dir

# Working with the file tree

<code>pwd</code>	print working directory
<code>ls [directory]</code>	list directory contents
<code>cd [directory]</code>	change directory
<code>.</code>	current Directory
<code>..</code>	parent Directory
<code>mkdir [directory]</code>	make directory
<code>touch [file]</code>	creates a file
<code>rm [file]</code>	removes a file
<code>rmdir [directory]</code>	removes a directory
<code>mv [SOURCE] [DESTINATION]</code>	Move/rename a file
<code>cp [SOURCE] [DESTINATION]</code>	Copy files and directories

# Help, I don't know what <cmd> does

- ❖ `man <cmd>`
  - Shows the manual
- ❖ Or just google it (<https://man7.org/linux/man-pages/man1/diff.1.html>)
  - May be easier to read
- ❖ Try find out what these do:
  - `Cat`, `head`, `echo`, `wc`, `diff`

# How to edit files from CLI?

## ❖ Emacs, vim, etc

- Steep learning curve
  - Type “vi test.txt”, using vim try to write “hello world” and then save + close the file
    - Hint: google “how to exit from vim”
- Once you master it, has same (or more) functionality as popular GUIs
  - Find and replace, go to definition, etc
  - Assignment 1: You have to learn how to do this

## ❖ Why?

- If you ssh into a server, you may not have a GUI.

## ❖ If you ever get stuck, google is your friend!

- “Emacs how to \_\_”
- <https://www.gnu.org/software/emacs/refcards/pdf/refcard.pdf>



# Shell variables

- ❖ Oftentimes need variables outside of programs
  - Create variables that only exist on your shell/CLI
- ❖ `pi=3.14, echo $pi`
  - Assignment: `variable = value`
  - Usage: `$variable`
- ❖ Why?
  - If you run a program, e.g. `ls`, where is the file located? What code should be run?
  - Look at the `PATH` variable (`$PATH`)
  - Try: find the location of `ls`

# Environment variables

- ❖ Environment variables are a set of variables that the current shell and any child processes of that shell will have access to
  - Look at all of your environment variables with `printenv`
- ❖ These become very important in large projects
- ❖ Add *`export PATH="/usr/local/cs/bin:$PATH"`* to your `~/.bash_profile` or the `~/.profile` file.
- ❖ *`export PATH=$PATH:/some/directory/to/append`*
- ❖ What does this command do?
  - Export will allow `PATH` to be seen by other processes
  - Use colon to keep original path variables (`$PATH`) and add new directory too

# Tips for emacs (assignment 1)

- ❖ Use the cheatsheet <https://www.gnu.org/software/emacs/refcards/pdf/refcard.pdf>
- ❖ C-<key> = control + a
- ❖ M-<key> = meta + key
  - Meta is usually alt
- ❖ Commands I find most useful
  - emacs <filename>
  - C-x C-s (save file)
  - esc esc esc (if you mess up. E.g. you pressed C-h instead of C-x)
  - C-v or M-v (scroll one screen down/up)
- ❖ Due next week, 12 Oct

Questions?