

# A Quick Unix Intro

Ay 120 Fall 2013

Jason Wang

# Poll

- Who has used UNIX or Linux before?
- Who has used the UNIX command line or terminal?

What is UNIX?

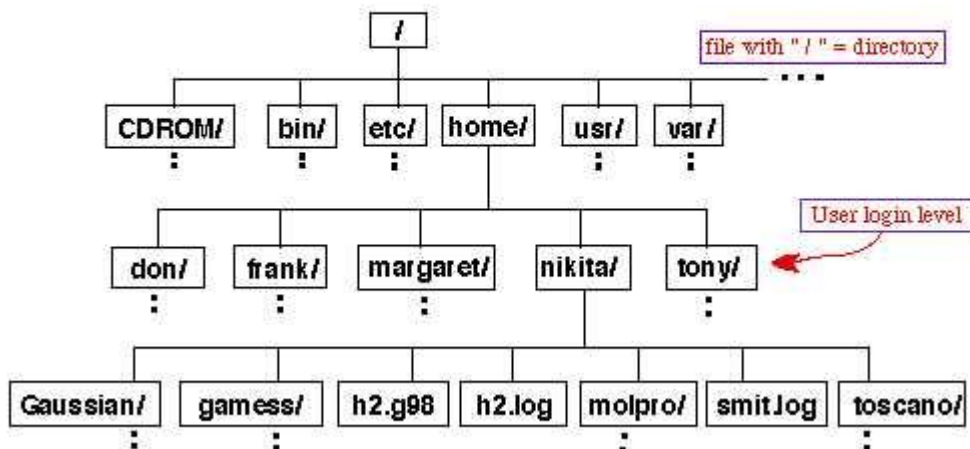
# What is UNIX?

- Short Answer: An Operating System
- A little longer answer: It provides an interface between the user and hardware via the UNIX command line
  - The kernel: Does all the hard work and behind the scenes magic
  - The shell: Provides a text interface to interact with the kernel
- UNIX isn't Linux
  - Linux and many other operating systems (including Mac OS) are built off of UNIX
- What is this openSUSE we have on all the computer then?
  - A UNIX-based operating system that provides extra features like a desktop, support for applications like Firefox, and pretty colors
- So why use the command line?
  - Powerful
  - It sometimes is the only option
  - Uniform across all UNIX-based systems
  - We will be using the command line to run python

# The UNIX Filesystem

- Structured like a tree
- / is the path to the root node
- /home/jwang is my home directory
  - When I log in, I start here (i.e. is my initial working directory)
- / at the end of a name denotes it is a directory
- Working directory denotes where you are in the filesystem currently
- Absolute vs. Relative file paths
  - Absolute file paths start with "/" and tells you how to get to a file or directory from the start of the tree
  - Relative file paths don't start with "/" and tells you how to get to a file or directory from your current working directory
  - You can use whichever you like

## File Structure in Unix



# How to navigate the filesystem

- Open a Terminal window
  - You should see `/home/yourusername%`
    - This tells you what your current directory is
- Confused where you are still?
  - `pwd` – prints working directory
- What files and folders are in my current directory?
  - `ls` – lists folders and files
- Change directories
  - `cd [dirname]` moves to that directory
  - `cd ..` moves up one level
- Create/remove a folder in this directory
  - `mkdir [dirname] & rmdir [dirname]`
- On openSUSE, you can always refer to the graphical file explorer

# Dealing with files

- Move a file around (cut)

- `mv [sourcefilepath] [destfilepath]`

- Copy file

- `cp [sourcefilepath] [destfilepath]`

- Delete a file

- `rm [filepath]`

- A quick way to read the contents of a file

- `less [filepath]`

- Usual keyboard buttons to navigate, press 'q' to exit

- Editing a file

- The first lab will instruct you to use `emacs`

- `emacs [filepath]` to open the file

- When in `emacs`, press `ctrl-x ctrl-c` to quit out

- Other text editors if you are adventurous: `vim` (what I use; a little harder to get used to), `nano`, `pico`

- Note: there are many text editors out there and the internet enjoys debating over which is best

- <http://xkcd.com/378/>

# Remotely accessing a computer

- `ssh` allows you to login to another computer
- If you're currently on a UG Lab computer
  - `ssh [username]@computername`
- Outside of the UG Lab, most computers are blocked by a firewall
  - `ssh [username]@ugastro.berkeley.edu`
  - Connects to `aequarius`
    - Doesn't have python libraries that you likely need
  - In general, I recommend you work here in the lab
- Transferring files
  - Your home directory (i.e. files) are available on any UG Lab computer
  - To transfer to a computer outside of the UG Lab, use `sftp [username]@ugastro.berkeley.edu`
  - Can use `cd/lis` to navigate directories
  - To copy a file from UG Lab: `get [sourcefilepath] [destfilepath]`
    - This is the same format as `cp` except the source filepath refers to the path on the remote machine and the destination filepath refers to the path on your machine
  - `put` does the opposite of `get`



# Some More Useful Commands

- `grep` lets you search text inside a file and outputs each line that contains the text you searched for
- `find -name '[searchwords]'` lets you search for filenames and directory names that are nested inside your current working directory
- `>>` redirects output to a file instead of displaying it on the command line
  - `>` does the same except clears the file first so no previous content in the file is preserved
  - Example: `ls > output.txt` writes the directory listing to the file `output.txt`
- `|` allows you to redirect the output of the command before the `|` to be the input of the command after the `|`
  - How can this be useful? See the next bullet
- `kill [PID]` allows you to terminate processes manually
  - How do you find the process identification number (PID)? `ps u | grep [processname]`
    - `ps` lists all of your running processes and displays them in a user-readable format (`u`) and passes the output to `grep` which will search the output for the process name and list out each process that has that name. Note: the second column in the output is the PID

# Helpful Hints

- When typing a filename, pressing 'tab' will autocomplete it as much as possible
  - Good for long filenames
- The up and down arrows allow you to cycle through previously typed commands
- `man [commandname]` gives you the documentation on that command
- Each command generally has a lot of options to modify its behavior which you can look up with `man`
- What you're trying to do in the command line is likely solved by someone on the internet so search away!
  - Usually better than using `man` to figure out how to do a particular task
  - Caveat: make sure you understand what each command does before typing it in (i.e. `man` it first!)

# Some online resources

- Check out the two UNIX tutorials on the course webpage for a more in-depth guide:  
<https://sites.google.com/site/ay120fall2013/home/tutorials-and-primers>
- Emacs cheat sheet for things you might want to do in emacs  
[http://www.rgrjr.com/emacs/emacs\\_cheat.html](http://www.rgrjr.com/emacs/emacs_cheat.html)
- Bonus: For those of you that think you have the hang of UNIX and want to learn some cool things you can do with the command line  
<http://www.commandlinefu.com/commands/browse/sort-by-votes>