CSE 374 Lecture 2

Notes

Office Hours:

Wednesdays, 2-3**am, PDT**

Mohit

Wednesdays, 5:30-6:30pm

Joyce

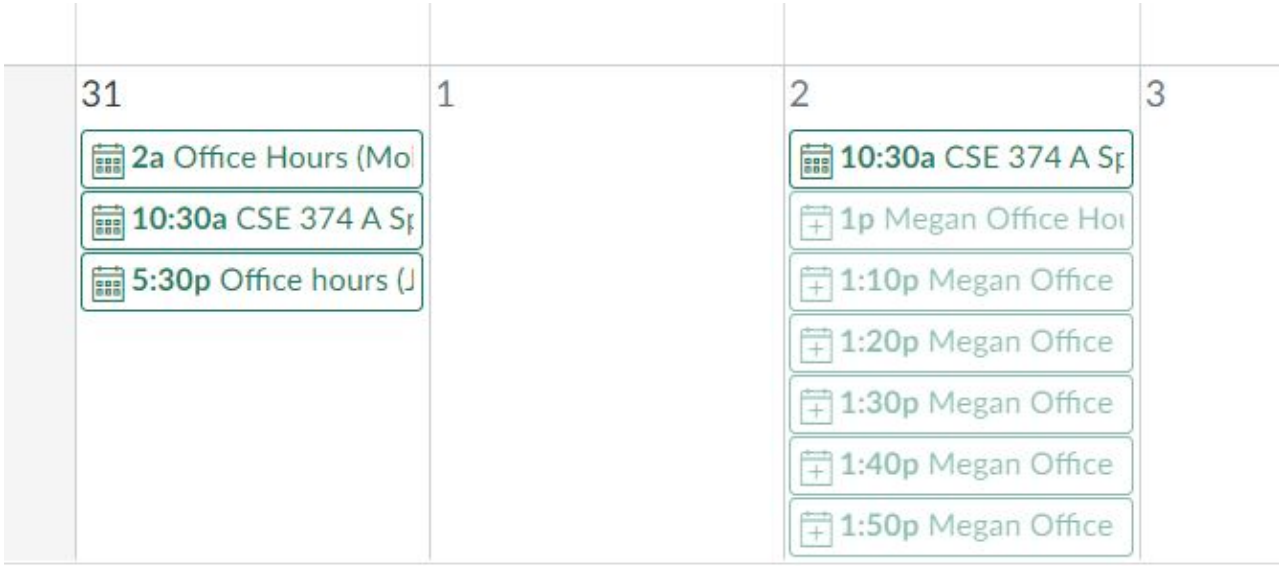
Thursdays, 1-2pm

Nick

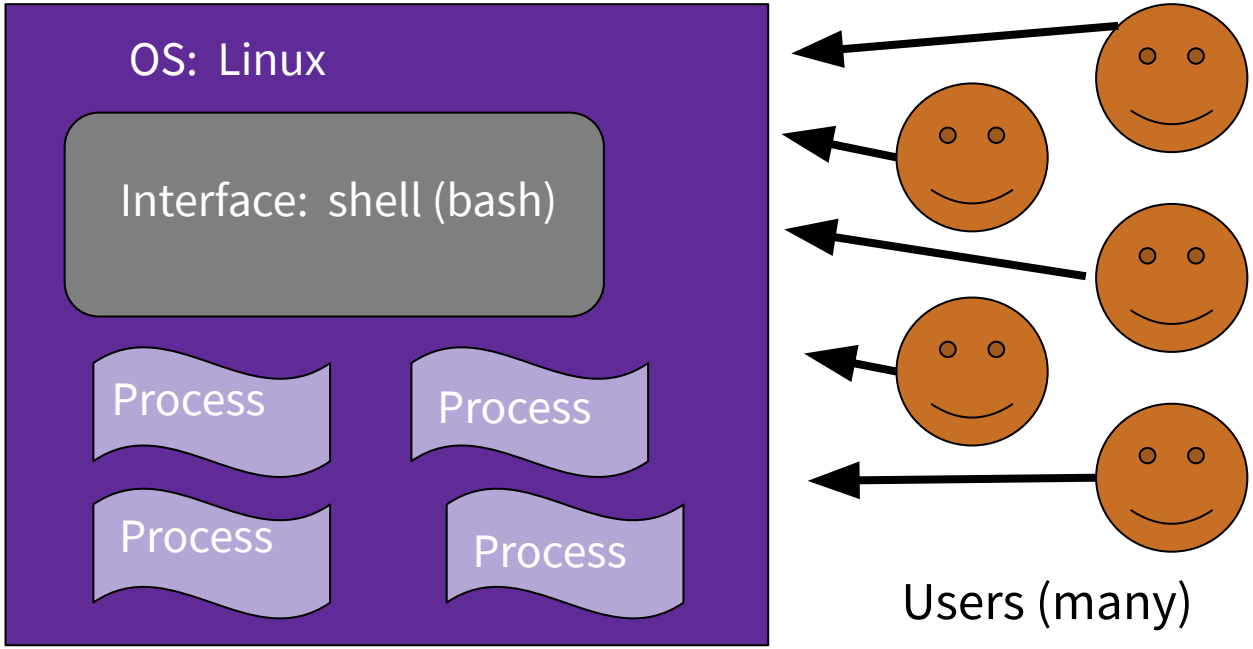
Fridays, tbd

Leah

Also,

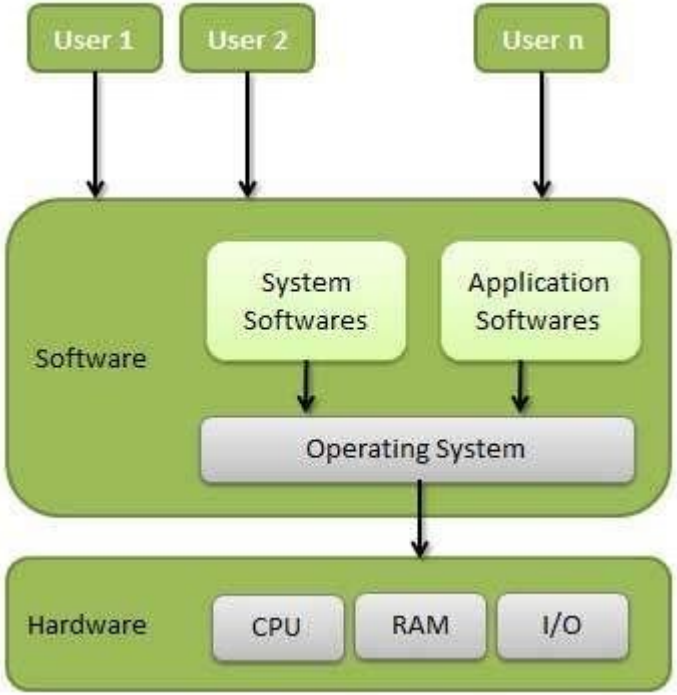


Computer Model



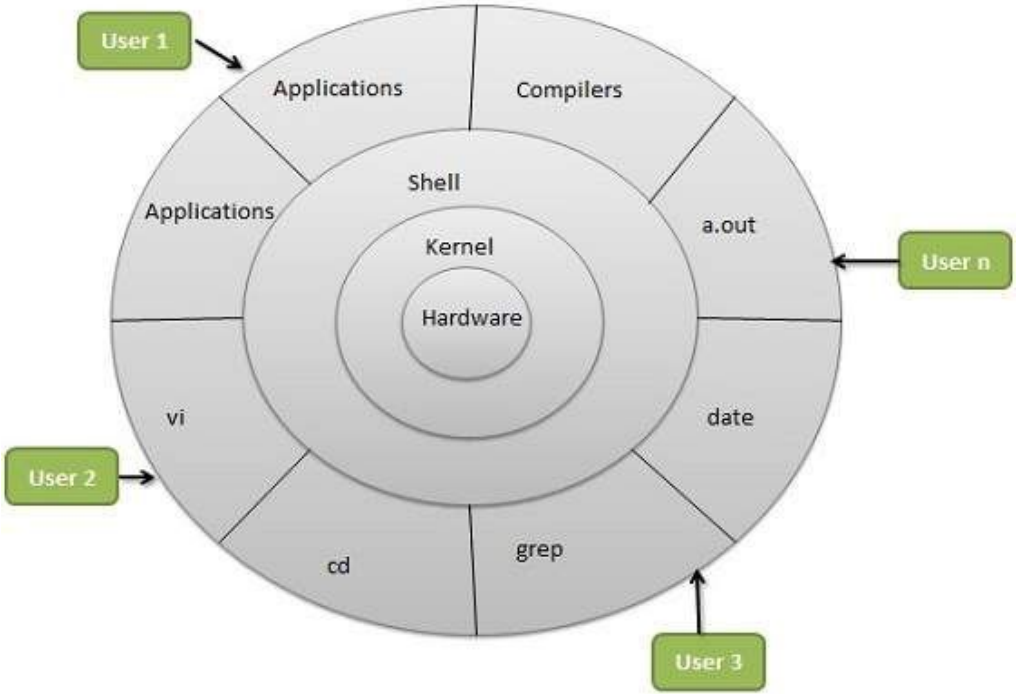
- One OS (CentOs) controls the computer.
- One filesystem stores data.
- Many processes are run. (A program runs one or many processes.)
- A shell is one process that allows for command line interface.
- Many users

What is the OS?



- Memory Management
- Processor Management
- Device Management
- File Management
- Security
- Control over system performance
- Job accounting
- Error detecting aids
- Coordination between other software and users

Linux Model



Linux -
Portable; multi-user

Includes

- Hardware layer (drivers, etc.)
- Kernel (does all the hardware interaction)
- Shell (provides user friendly interface to kernel)
- Processers (various programs)
- Users - multiple users run processes

http://www.tldp.org/LDP/intro-linux/html/chap_01.html

Getting Started with Linux (CentOS)

Use a virtual machine

Or

Log on to Klaatu

https://courses.cs.washington.edu/courses/cse374/21sp/resources/linux.html

Log-in and get a ‘shell’

- Shell - text based interface
- Specifically ‘bash’

Everyone should have a klaatu account - send email to cse374-staff if you can not access yours.

Linux & Shells

Text is efficient - typing is fast, and there aren't big image objects to pass around

Scripting makes it easy to automate text based interfaces

Linux *does* have a graphical interface

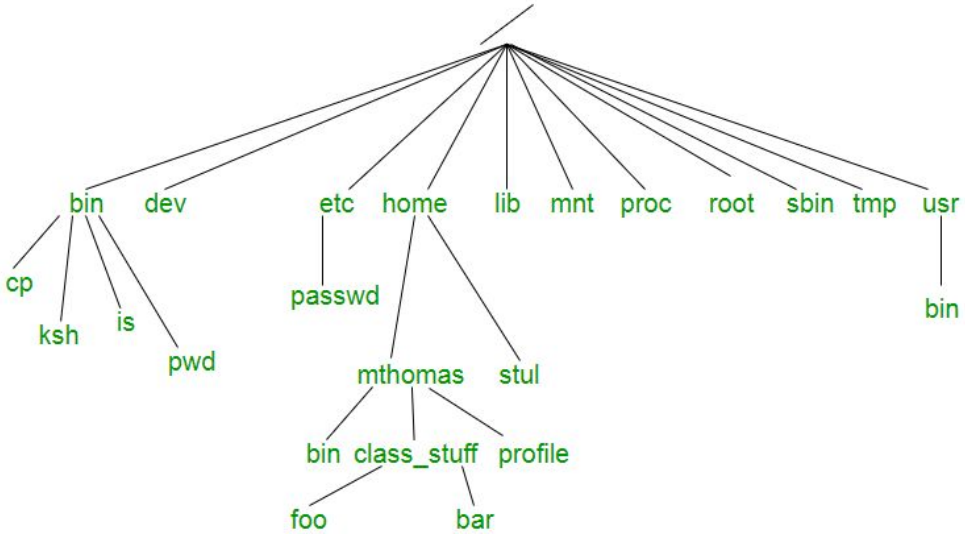
Windows and MacOS *do* have shell interfaces

Most power users use BOTH

You could use any distribution of Linux that is up-to-date. Using CentOS through CSE ensures consistency.

(What a distribution? Something like a 'flavor', or a branded implementations. Distributions vary somewhat.

File Systems



- File systems are trees
- (or directed acyclic graphs)
- A file (or directory) is specified by its path from the top (‘/’)
- Can be specified absolutely (entire path),
- Relatively (from current location)
  - This directory ‘./’
  - One directory up ‘../’
- You have access to your ‘home’ directory (‘~’)

Also true on Windows, btw, although the structure and some notation is different.
Demo - whoami, pwd, ls, mkdir, cd, cp, mv, rm less,more,text editors
http://www.tldp.org/LDP/intro-linux/html/sect_03_01.html

Processes & the Shell

"On a UNIX system, everything is a file; if something is not a file, it is a process."

Shell essentially runs programs, or processes. Shell *is* a process, and has a state.

Usually launch a process, and return to shell when done.

Each process has own memory stream and I/O

Stdin (keyboard), stdout (console), stderr

Many processes have options

‘&’ runs process in the background

‘fg’, ‘bg’, top, kill

Control where processes run

Step through a script with built-in ‘source’

Can redirect input and output (‘<’, ‘>’)

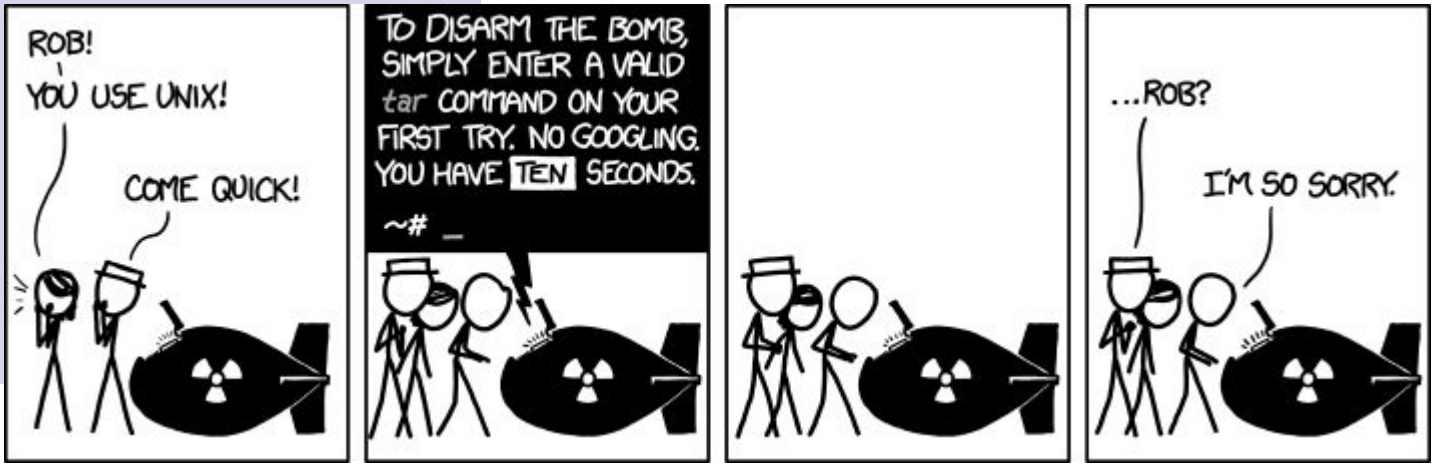
Getting Help

Most commands: ‘man ls’

Also “--help”

Look for keyword: ‘man -k’

http://www.tldp.org/LDP/intro-linux/html/sect_02_03.html



Bash Language

- Bash acts as a language interpreter
  - Commands are subroutines with arguments
  - Bash interprets the arguments & calls subroutine
  - Bash also has its own variables and logic

BASH applies its own processing to the I/O text - ‘globbing’

Shell Behavior

All redirection & string expansion or substitutions are done by the shell, before the command.

Command only sees resulting I/O streams.

Special Characters

- Directory Shortcuts
  - ~uname or ~
  - ./ or ../
- Wildcards - *Globbing*
  - 0 or more chars: *
  - Exactly 1 char: ?
  - Specified chars: [a-f]

History, or ‘!’

Special Characters

! > < & | * ~ [] “ ‘ ` $ /

What do they all mean?

\ is escape character

Would substitute things like $VAR

“string”

‘string’

Suppresses substitutions