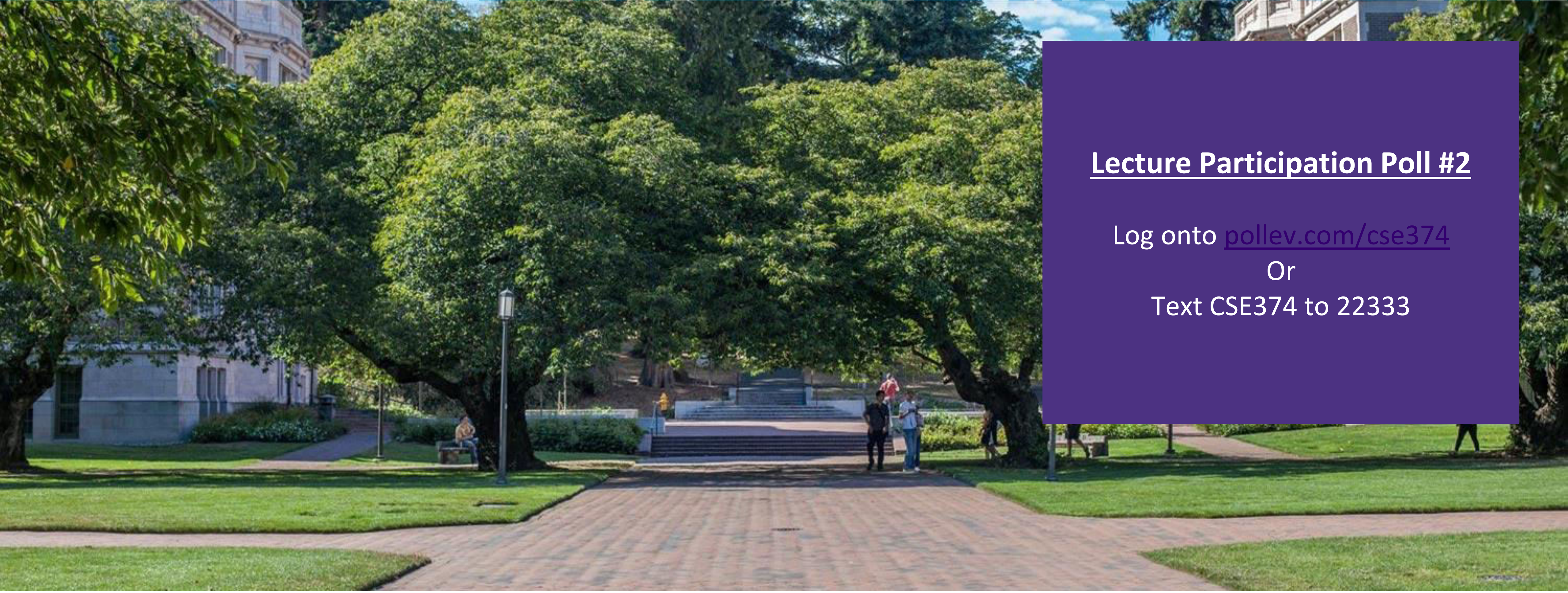


Lecture Participation Poll #2

Log onto pollev.com/cse374
Or
Text CSE374 to 22333

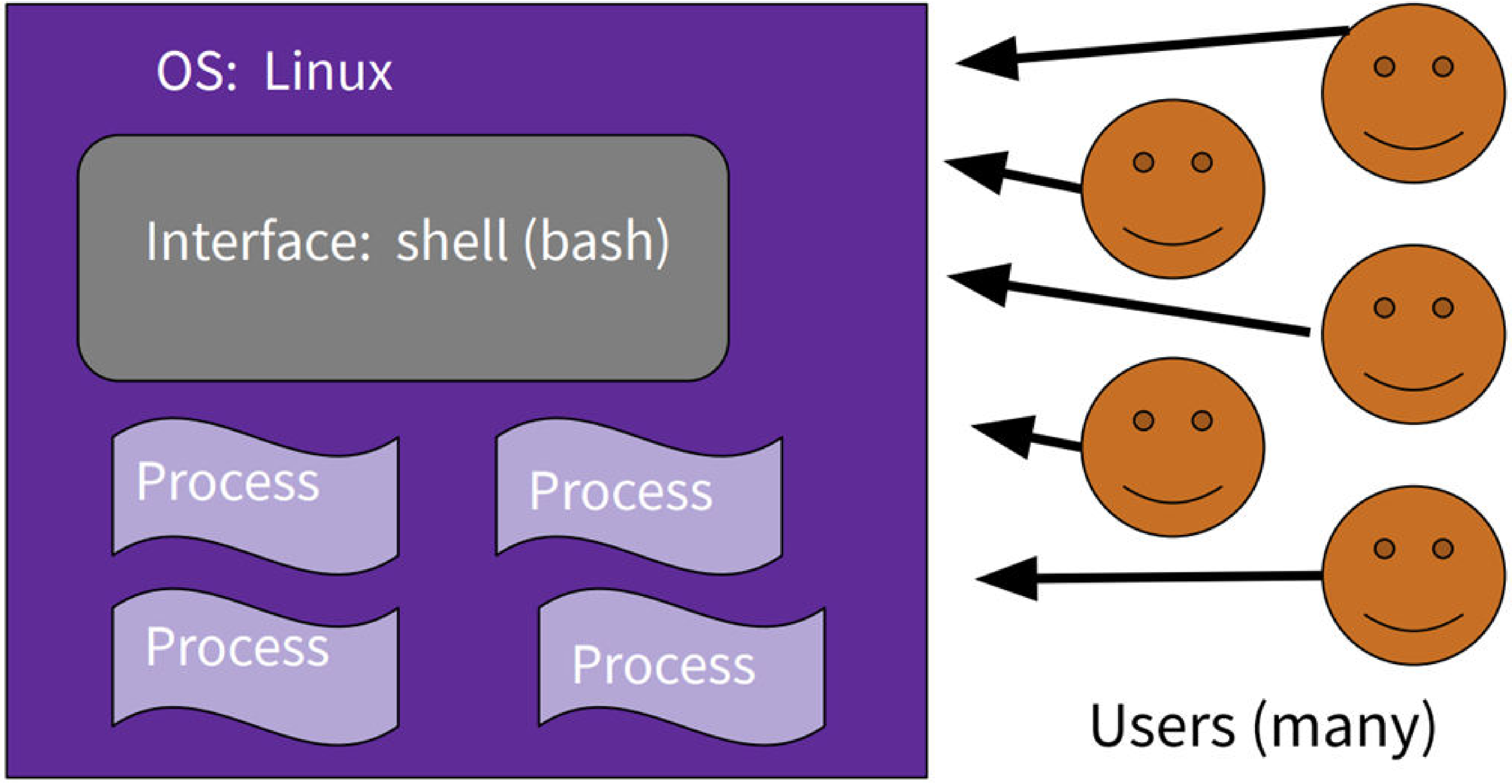
Lecture 2: Linux Shell & Files

CSE 374: Intermediate Programming Concepts and Tools

Administrivia

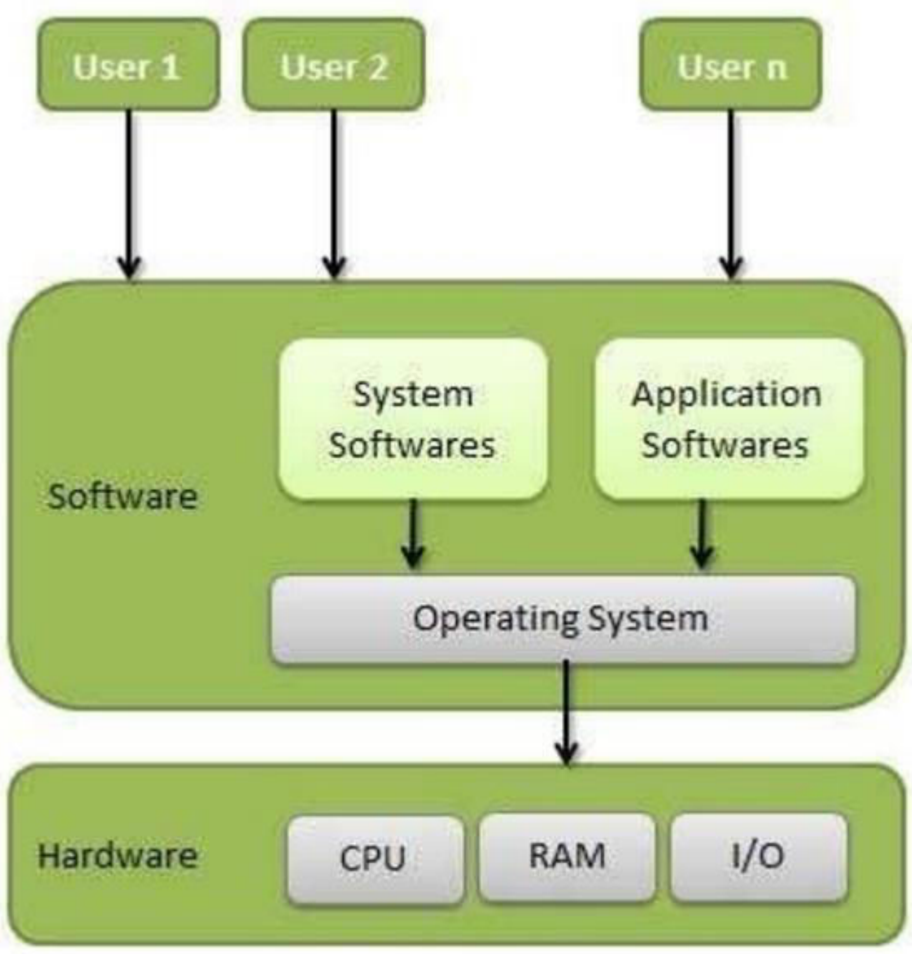
- HW 1 will release Monday
- Class webpage coming later today
- Class discussion board available
- Linux accounts will be available later this afternoon
  - Username = uwnetid
  - Password = tempPassword
- Meet some of your TAs!
  - Dixon
  - Andres
  - Tom
  - Leah

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

Bash Language

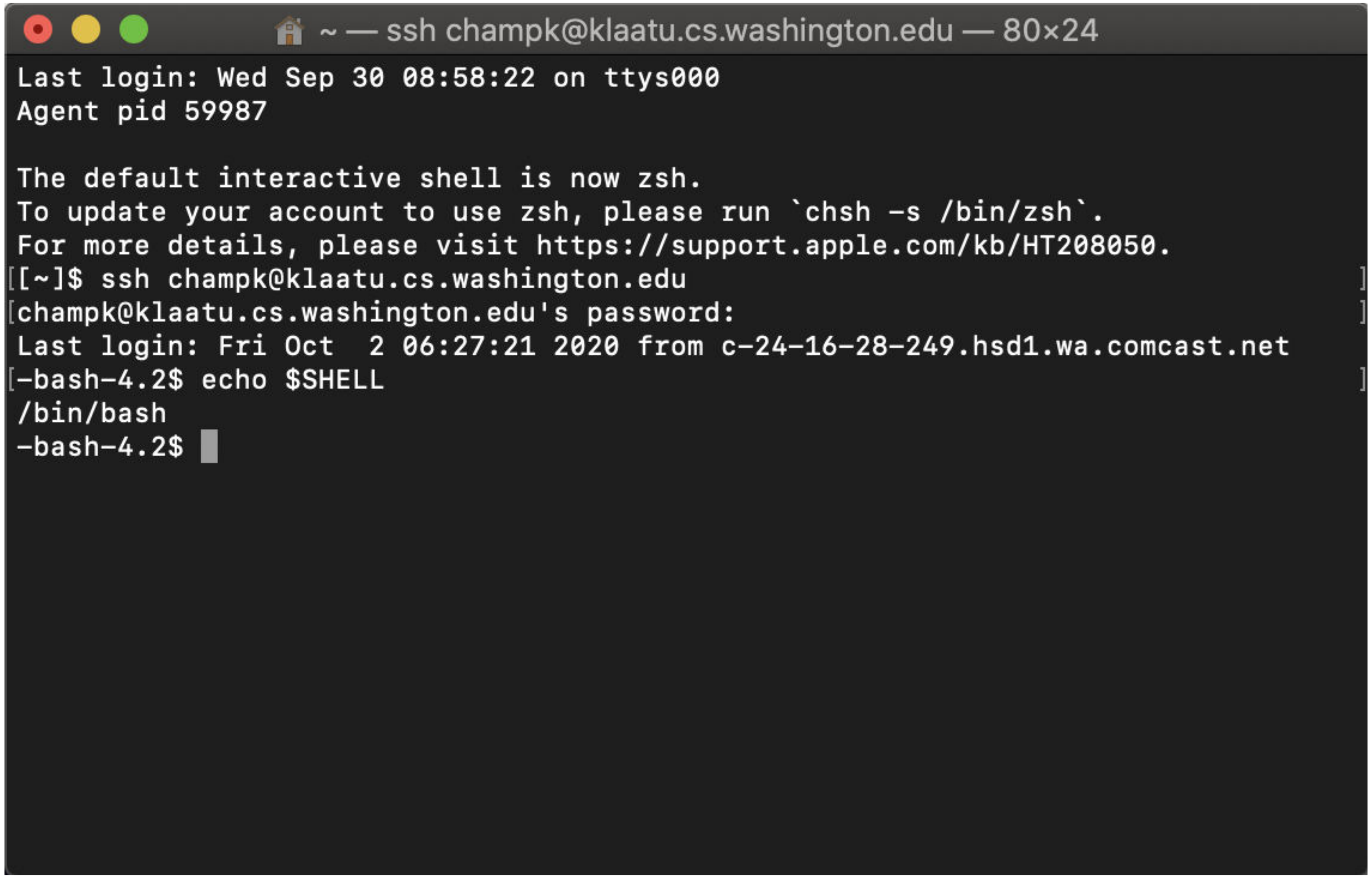
Bash acts as a language interpreter

- commands are subroutines with arguments
- bash interprets the arguments and calls subroutine
- bash has its own variables and logic

Bash	Java
Interpreted	Compiled
Esoteric variable access	Highly structured & strongly typed
everything is a string	Strings have library processing
easy access to files and programs	Data structures and libraries
good for quick & interactive programs	good for large complex programs

Meet the Linux Shell

- Text based interface for Linux operating system
- We will be using the "Bash" shell
  - There are different versions, but for this course we will only be using bash
- Use echo $SHELL to check which shell you are using
- Bash in a unix shell and command language that is the default login shell for most Linux and MacOS
- Interpreted, not compiled
  - You're on your own when things go wrong



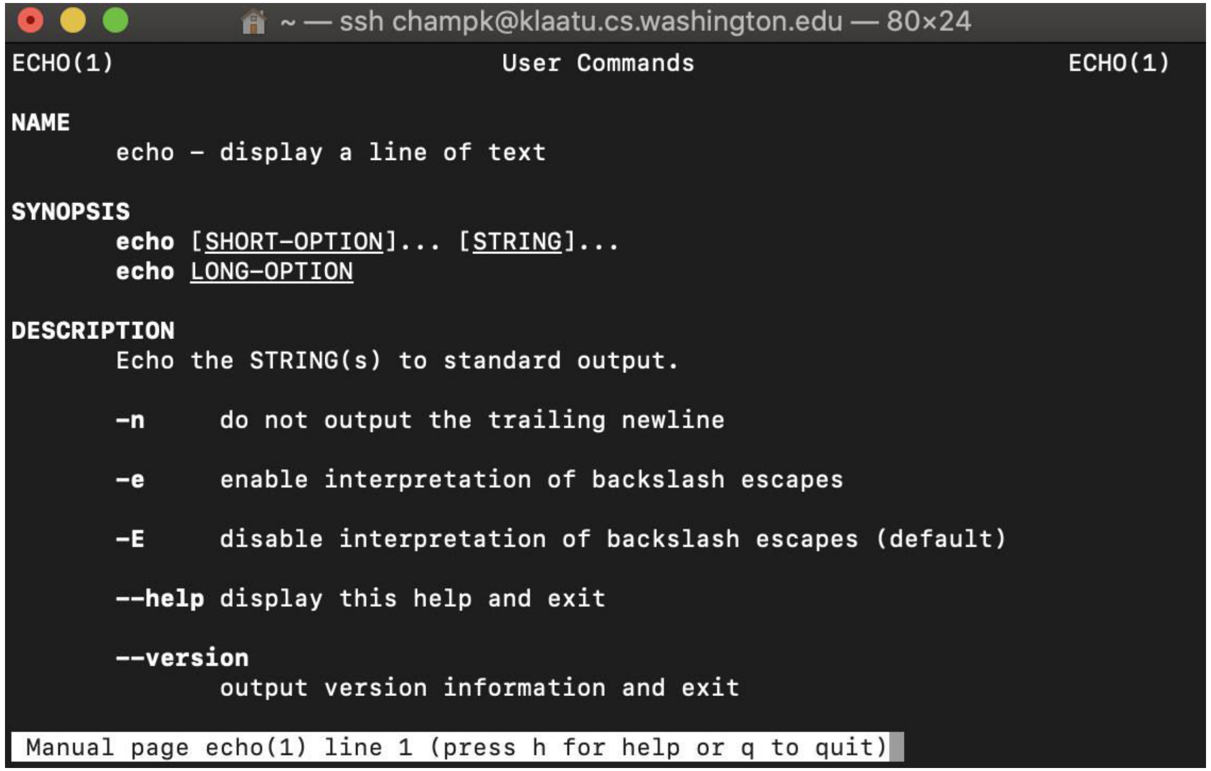
```
Last login: Wed Sep 30 08:58:22 on ttys000
Agent pid 59987

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[~]$ ssh champk@klaatu.cs.washington.edu
champk@klaatu.cs.washington.edu's password:
Last login: Fri Oct  2 06:27:21 2020 from c-24-16-28-249.hsd1.wa.comcast.net
-bash-4.2$ echo $SHELL
/bin/bash
-bash-4.2$ 
```

Local MacOS terminal connecting to remote Linux machine

Commands in the Shell

- The shell is a text-based interface that takes commands instead of clicks
- Commands are pre-existing programs
  - <command name> <options> <input || output>
- To learn about an individual command use "man"
  - <command name> man
  - Short for "manual page"
  - Can also use the --help option



```
ECHO(1)                          User Commands                         ECHO(1)

NAME
       echo - display a line of text

SYNOPSIS
       echo [SHORT-OPTION]... [STRING]...
       echo LONG-OPTION

DESCRIPTION
       Echo the STRING(s) to standard output.

       -n     do not output the trailing newline

       -e     enable interpretation of backslash escapes

       -E     disable interpretation of backslash escapes (default)

       --help display this help and exit

       --version
              output version information and exit

 Manual page echo(1) line 1 (press h for help or q to quit)
```

echo man page

Shell Interaction Basics

1. Open the terminal application on your local computer
2. Connect to Klaatu Linux server with
3. ssh <username>@klaatu.cs.washington.edu
4. Enter in your password, you will not see characters as you type

Basic Interactions:

- You can use copy and paste with with your usual short cuts
- You can navigate through your executed commands by using the up and down arrows
  - Convenient way to rerun commands or to fix small errors in previous command
- The history command will print the commands you've used this session to the terminal

Running Programs

- You can run a program by typing its path into the terminal
- Some folders are globally visible, so you only need the program's name
  - /bin/ is globally visible because it is in the PATH shell variable
- To run a program in the current directory you need to give the path
  - ./local_program
  - Running local_program by itself will not work because it's not globally visible
- All commands are bash files that are executed when you hit "enter" on a terminal line
  - You can write and execute your own! More on that later

Processes in the Shell

- Programs running in the shell are called “processes”
  - We refer to the code/instructions as the “program”
  - We can run a given program many times, creating many processes
  - Terminal can only run one process in the foreground at a time
    - Use the ”`&`” special character to launch a process in the background
    - EX: `emacs &`
- Bash Shell has many built in programs
  - Commands like `cd` and `ls`
- Processes have Input and Outputs
  - Inputs come in two main forms: arguments and stdin
    - Arguments are strings separated by spaces given after the command
  - EX: `cp my/src dest/folder`
      - Arguments: “`my/src`” and “`dest/folder`”
      - Arguments with spaces need to be wrapped in quotes EX: `echo “hello world”`
    - Stdin or Standard Input is a stream that the user enters into the terminal
  - Outputs can be stdout, stderr or a directed to an output file
  - All redirections & string expansions or substitutions are done by the shell before the command

Useful Commands

Command	Operation	Example
`ls`	See folder contents	`ls -l`
`cd <folderName>`	Move into given folder	`cs Downloads`
`cp <source> <destination>`	Make a copy of given file in given destination	`cp file.txt myDir/`
`mv <oldName> <newname>`	Rename or move given existing file to given name/destination	`mv fil.txt file.txt`
`cat <fileName>`	Print file contents to terminal window	`cat file.txt`
`touch <filename>`	Create empty file with given name	`touch file.txt`
`echo <string>`	Print given string to terminal window	`echo “hello world”`
`pwd`	Print working directory	`pwd`
`mkdir <directoryName>`	Create an empty directory at location specified	`mkdir ~/newDir`
`exit`	Exit the shell	`exit`

Other Useful Commands

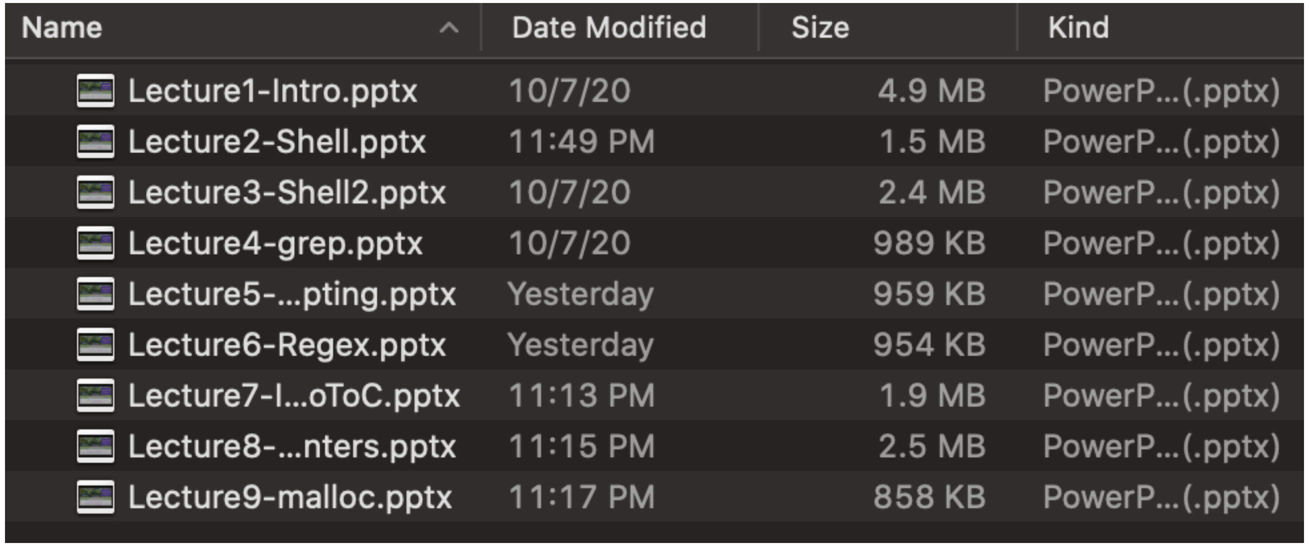
Command	Operation	Example
`pico <fileName>`	Create or edit files	`pico filename`
`echo <text>`	Print text	`echo hello world`
`pwd`	Print working directory's absolute path	`pwd`
`touch <filename>`	Create empty file	`touch filename`
`mkdir`	Create empty directory	`mkdir`
`find –name <filename>`	Search for file	
`exit`	Exit the shell	

Linux Demo

Recorded Demo from 374 Sp 20 Instructor Megan Hazen

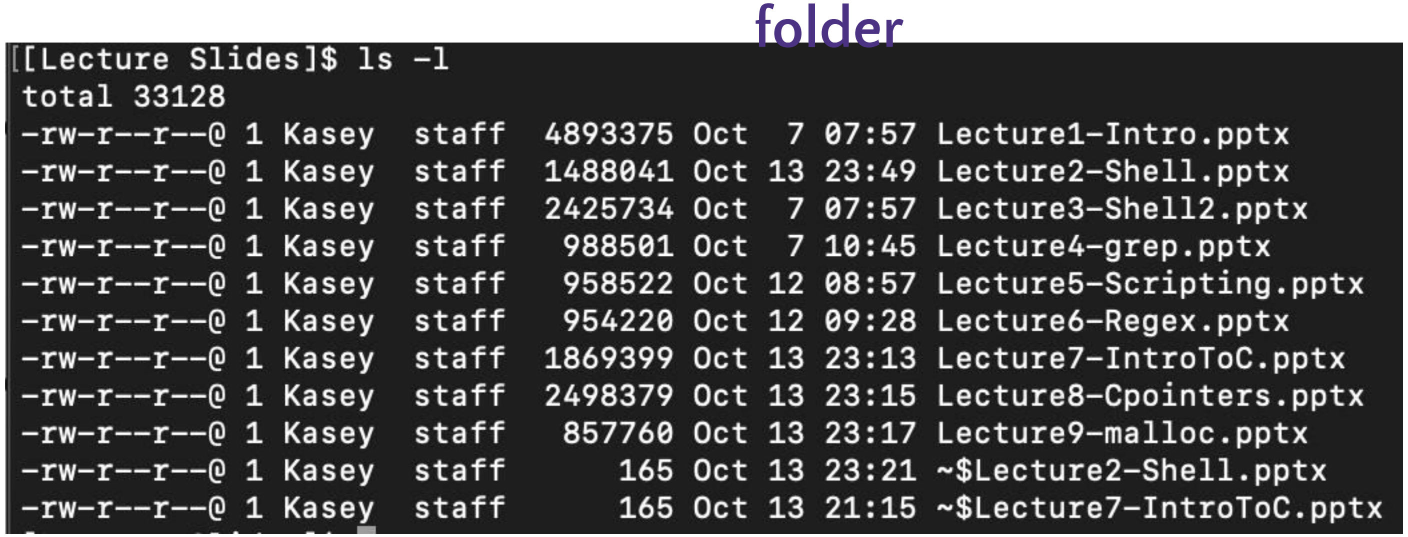
Files

- A collection of data used for long term storage
  - Stored on a hard drive
  - Hard drive is the physical portion of a computer that stores large amounts of data sits outside the CPU
- Files have…
  - Name
    - Unique string within the folder
  - Type
    - Indicated by the extension at the end of the name
  - Content
    - Data contained within the file
  - Location
    - Folder trail from drive to name
    - “breadcrumb”



Name	Date Modified	Size	Kind
Lecture1-Intro.pptx	10/7/20	4.9 MB	PowerP...(.pptx)
Lecture2-Shell.pptx	11:49 PM	1.5 MB	PowerP...(.pptx)
Lecture3-Shell2.pptx	10/7/20	2.4 MB	PowerP...(.pptx)
Lecture4-grep.pptx	10/7/20	989 KB	PowerP...(.pptx)
Lecture5-...pting.pptx	Yesterday	959 KB	PowerP...(.pptx)
Lecture6-Regex.pptx	Yesterday	954 KB	PowerP...(.pptx)
Lecture7-I...oToC.pptx	11:13 PM	1.9 MB	PowerP...(.pptx)
Lecture8-...nters.pptx	11:15 PM	2.5 MB	PowerP...(.pptx)
Lecture9-malloc.pptx	11:17 PM	858 KB	PowerP...(.pptx)

Finder GUI view of folder



```
[[Lecture Slides]$ ls -l
total 33128
-rw-r--r--@ 1 Kasey  staff  4893375 Oct  7 07:57 Lecture1-Intro.pptx
-rw-r--r--@ 1 Kasey  staff  1488041 Oct 13 23:49 Lecture2-Shell.pptx
-rw-r--r--@ 1 Kasey  staff  2425734 Oct  7 07:57 Lecture3-Shell2.pptx
-rw-r--r--@ 1 Kasey  staff   988501 Oct  7 10:45 Lecture4-grep.pptx
-rw-r--r--@ 1 Kasey  staff   958522 Oct 12 08:57 Lecture5-Scripting.pptx
-rw-r--r--@ 1 Kasey  staff   954220 Oct 12 09:28 Lecture6-Regex.pptx
-rw-r--r--@ 1 Kasey  staff  1869399 Oct 13 23:13 Lecture7-IntroToC.pptx
-rw-r--r--@ 1 Kasey  staff  2498379 Oct 13 23:15 Lecture8-Cpointers.pptx
-rw-r--r--@ 1 Kasey  staff   857760 Oct 13 23:17 Lecture9-malloc.pptx
-rw-r--r--@ 1 Kasey  staff      165 Oct 13 23:21 ~$Lecture2-Shell.pptx
-rw-r--r--@ 1 Kasey  staff      165 Oct 13 21:15 ~$Lecture7-IntroToC.pptx
```

ls –l view of folder

Linux File Permissions

Permission Groups

- **u** – Owner
- **g** – Group
- **o** – Others
- **a** – All users

Permission Types

- **r**– read – a user's ability to read the contents of the file.
- **w** – write – a user's capability to write or modify a file or directory.
- **x** – execute – a user's capability to execute a file or view the contents of a directory.

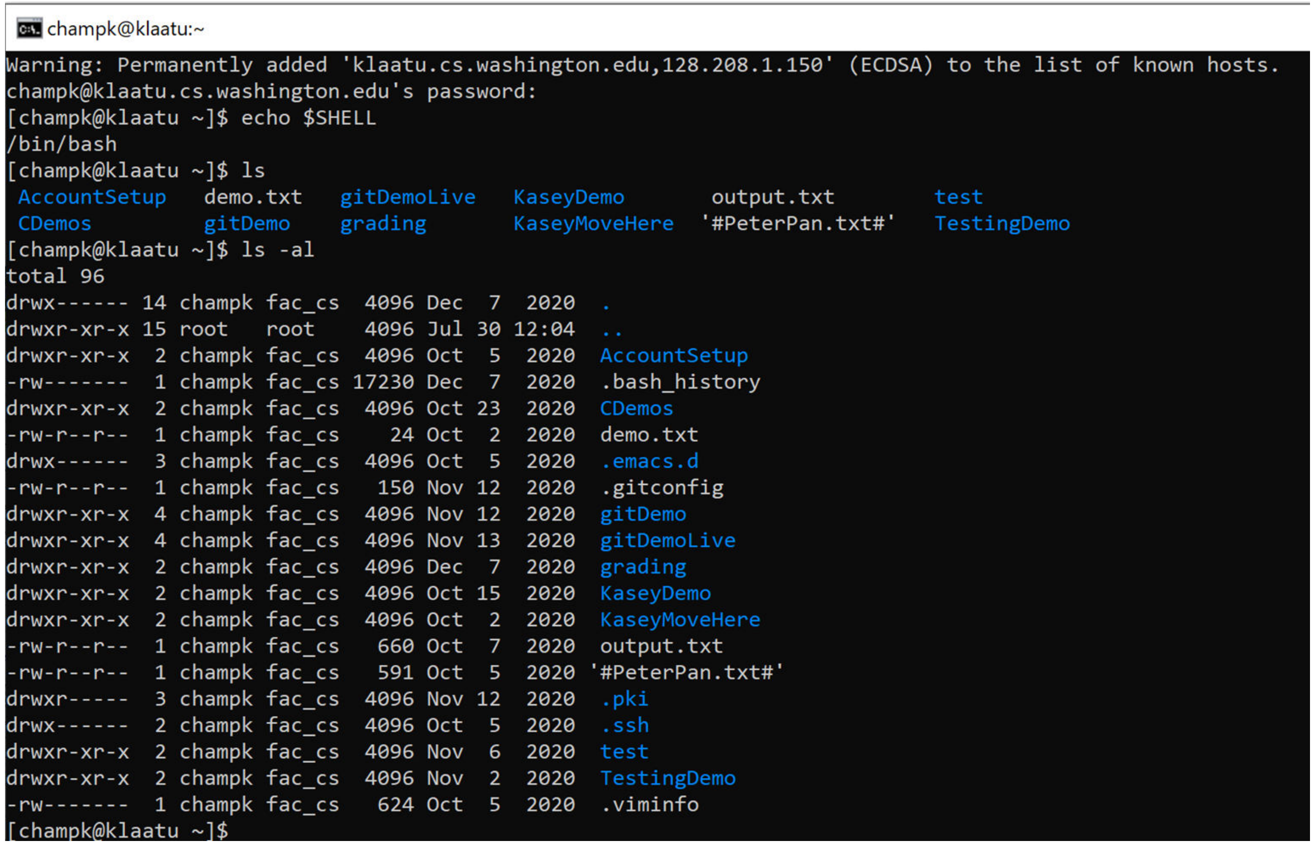
reading ls -l

- `_rw_rw_rw` = owner, group and all users have read & write permissions
- first character is either a - or a d : d means "directory", "-" means file

chmod <group>+||-<permission> <file>

- `chmod a-rw file1` : remove read and write permissions on file1 for all users
- `chmod a+rw file1` : add read and write permissions on file1 for all users

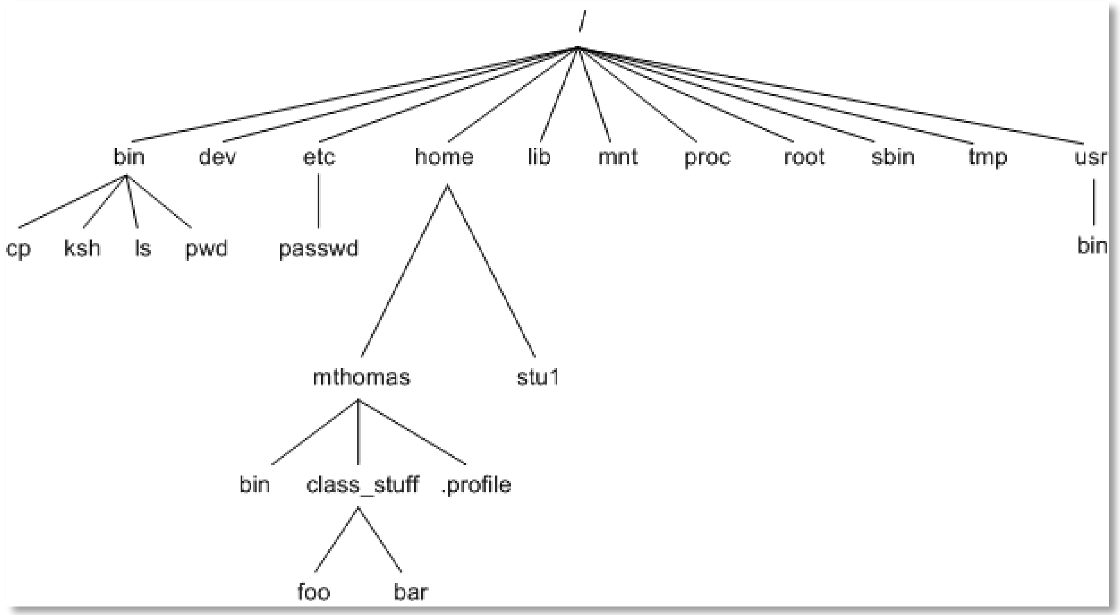
https://www.linux.com/training-tutorials/understanding-linux-file-permissions/



```
Warning: Permanently added 'klaatu.cs.washington.edu,128.208.1.150' (ECDSA) to the list of known hosts.
champk@klaatu.cs.washington.edu's password:
[champk@klaatu ~]$ echo $SHELL
/bin/bash
[champk@klaatu ~]$ ls
 AccountSetup    demo.txt   gitDemoLive   KaseyDemo        output.txt         test
 CDemos          gitDemo    grading       KaseyMoveHere   '#PeterPan.txt#'   TestingDemo
[champk@klaatu ~]$ ls -al
total 96
drwx------ 14 champk fac_cs  4096 Dec  7  2020  .
drwxr-xr-x 15 root   root    4096 Jul 30 12:04  ..
drwxr-xr-x  2 champk fac_cs  4096 Oct  5  2020  AccountSetup
-rw-------  1 champk fac_cs 17230 Dec  7  2020  .bash_history
drwxr-xr-x  2 champk fac_cs  4096 Oct 23  2020  CDemos
-rw-r--r--  1 champk fac_cs    24 Oct  2  2020  demo.txt
drwx------  3 champk fac_cs  4096 Oct  5  2020  .emacs.d
-rw-r--r--  1 champk fac_cs   150 Nov 12  2020  .gitconfig
drwxr-xr-x  4 champk fac_cs  4096 Nov 12  2020  gitDemo
drwxr-xr-x  4 champk fac_cs  4096 Nov 13  2020  gitDemoLive
drwxr-xr-x  2 champk fac_cs  4096 Dec  7  2020  grading
drwxr-xr-x  2 champk fac_cs  4096 Oct 15  2020  KaseyDemo
drwxr-xr-x  2 champk fac_cs  4096 Oct  2  2020  KaseyMoveHere
-rw-r--r--  1 champk fac_cs   660 Oct  7  2020  output.txt
-rw-r--r--  1 champk fac_cs   591 Oct  5  2020 '#PeterPan.txt#'
drwxr-----  3 champk fac_cs  4096 Nov 12  2020  .pki
drwx------  2 champk fac_cs  4096 Oct  5  2020  .ssh
drwxr-xr-x  2 champk fac_cs  4096 Nov  6  2020  test
drwxr-xr-x  2 champk fac_cs  4096 Nov  2  2020  TestingDemo
-rw-------  1 champk fac_cs   624 Oct  5  2020  .viminfo
[champk@klaatu ~]$
```

File System

- Files contain other files, branching out from the root "/" forming a tree-like hierarchy
- Files are located with a path of folders separated by "/" this is called the "file path"
- Paths starting with "/" are called absolute paths
  - Start searching from the root of the file system
  - EX: /usr/documents/myFiles/myFile.txt
- Paths that do NOT start with "/" are called relative paths
  - Starts searching from current directory
  - EX: myFiles/myFile.txt
- pwd command will print the current directory



Tree diagram of file structure

https://homepages.uc.edu/~thomam/Intro_Unix_Text/File_System.html

Demo: File Manipulation

Questions?

Lecture Participation Poll #2

Log onto pollev.com/cse374
Or
Text CSE374 to 22333