Rich's lesson module checklist

Last updated 10/13/2019

- ❑ Zoom recording named and published for previous lesson

- ❑ Slides posted
- ❑ Print out agenda slide and annotate page numbers

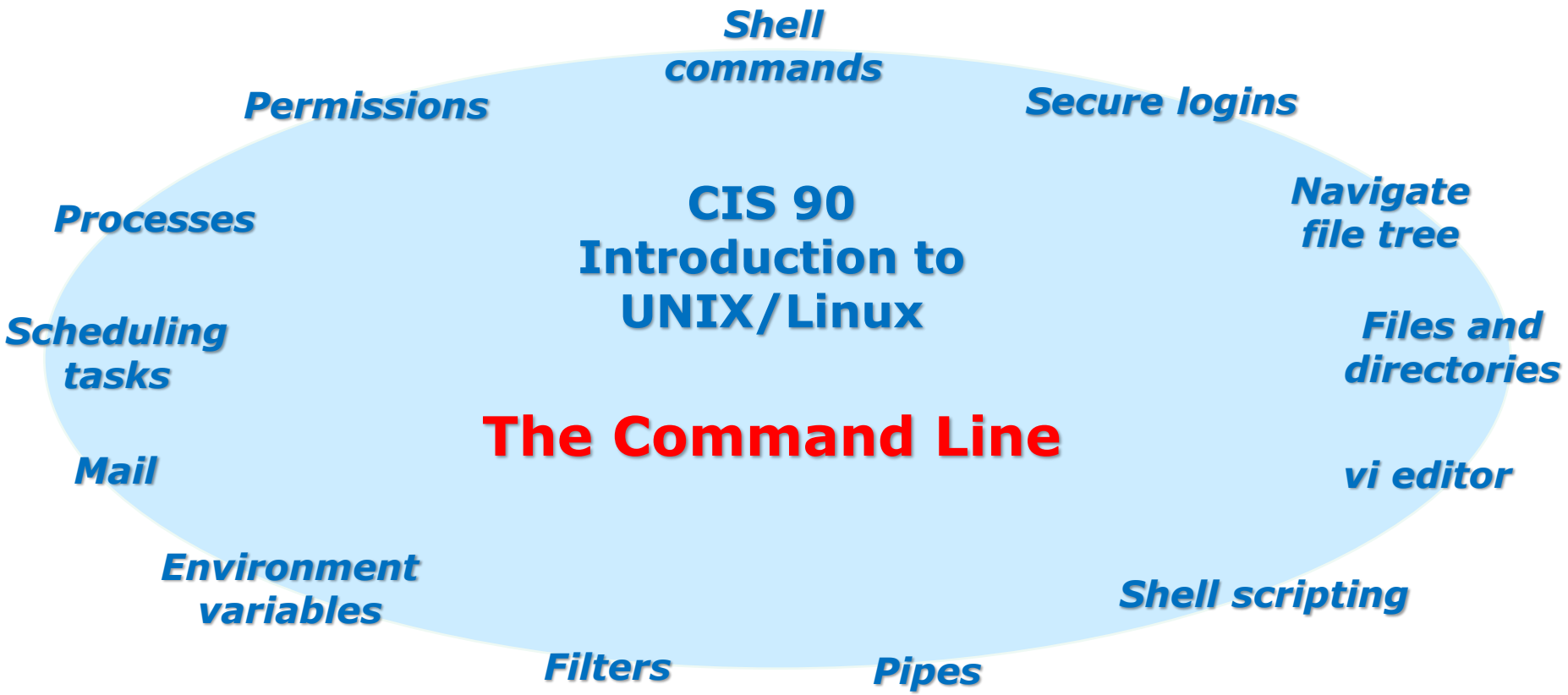
- ❑ 1st minute quiz
- ❑ Flash cards
- ❑ Calendar page updated

- ❑ Schedule lock of turnin directory and submit
  scripts/schedule-submit-locks
- ❑ Opus-II - hide script tested
- ❑ Update test Q21 for number of accounts (Sp18 /var "lost" two city files - FIX)
- ❑ Practice test available on Canvas at end of class
- ❑ P2 test system online and unlocked at end of class
- ❑ Flash cards, teams and timer script ready
- ❑ Clean up mysql database

- ❑ 9V backup battery for microphone
- ❑ Backup slides, CCC info, handouts on flash drive
- ❑ Key card for classroom door

- ❑ https://zoom.us

- ❑ Putty, slides, Chrome
- ❑ Enable/Disable attendee sharing
  ^ > Advanced Sharing Options > Only Host
- ❑ Enable/Disable attended annotations
  Share > More > Disable Attendee Sharing



Student Learner Outcomes

1. Navigate and manage the UNIX/Linux file system by viewing, copying, moving, renaming, creating, and removing files and directories.
2. Use the UNIX features of file redirection and pipelines to control the flow of data to and from various commands.
3. With the aid of online manual pages, execute UNIX system commands from either a keyboard or a shell script using correct command syntax.

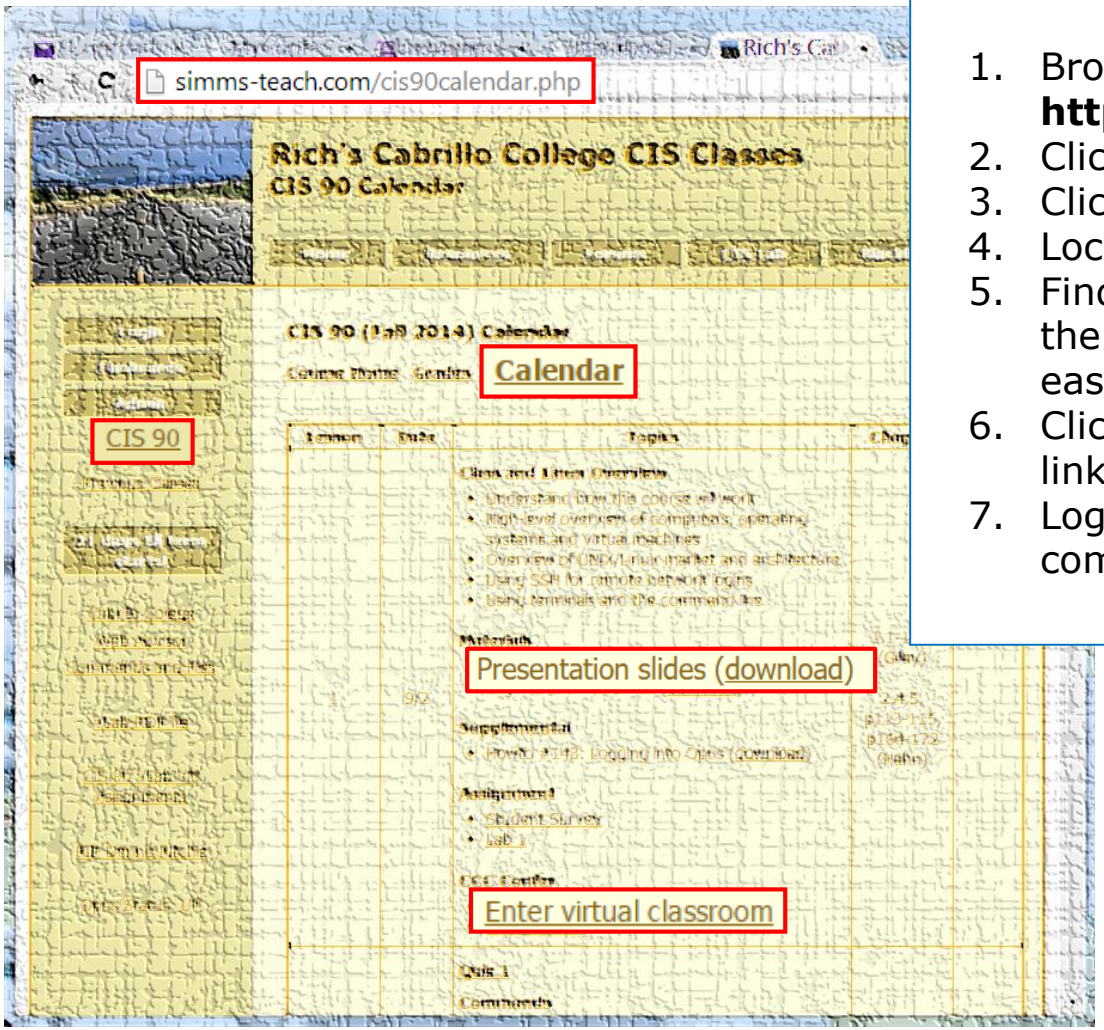
Introductions and Credits

Jim Griffin
- Created this Linux course
- Created Opus and the CIS VLab
- Jim's site: https://web.archive.org/web/20140209023942/http://cabrillo.edu/~jgriffin/

Rich Simms
- HP Alumnus
- Started teaching this course in 2008 when Jim went on sabbatical
- Rich's site: http://simms-teach.com

And thanks to:
- John Govsky for many teaching best practices: e.g. the First Minute quizzes, the online forum, and the point grading system. John's site: http://teacherjohn.com/
- Jaclyn Kostner for many webinar best practices: e.g. mug shot page.

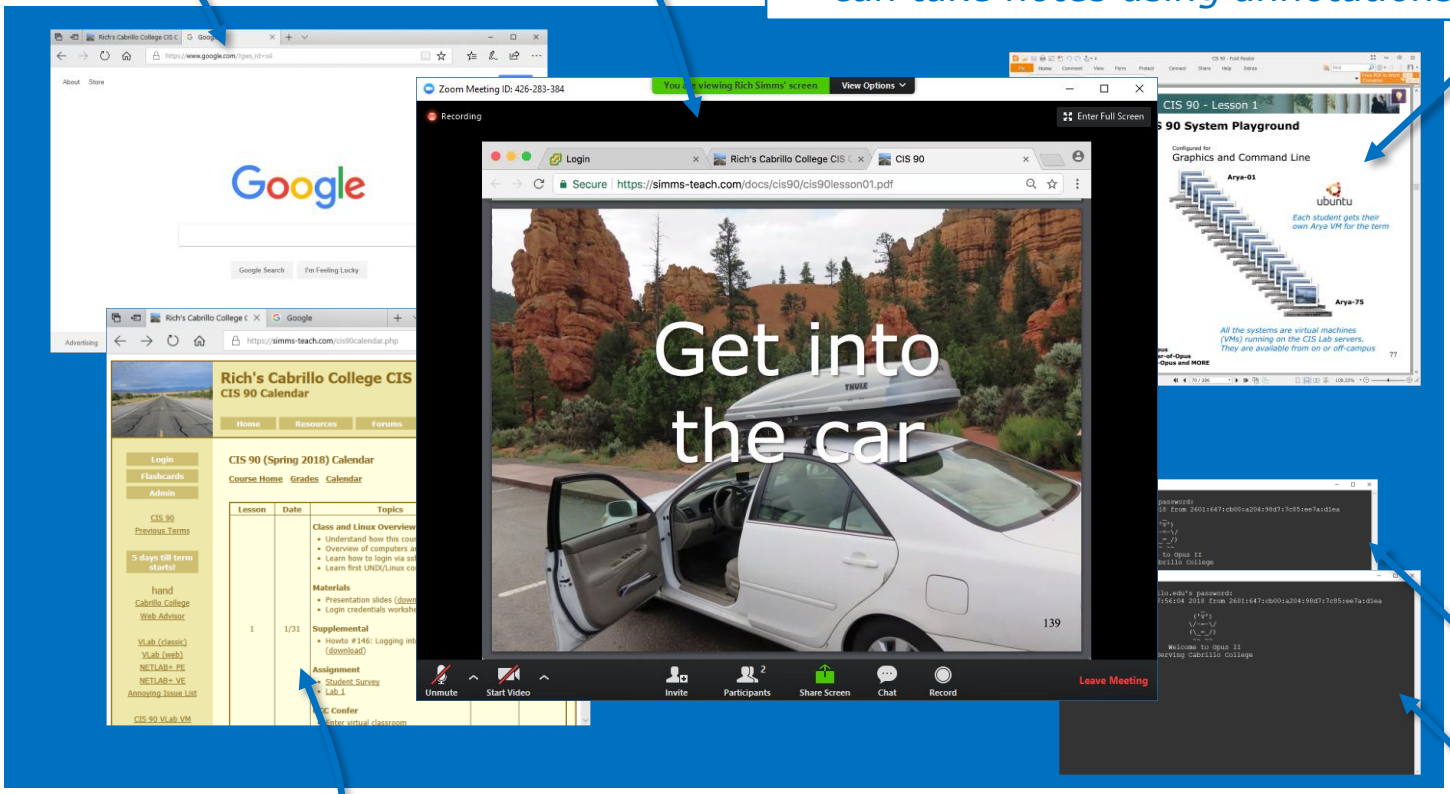
Student checklist - Before class starts



1. Browse to: **http://simms-teach.com**
2. Click the **CIS 90** link.
3. Click the **Calendar** link.
4. Locate today's lesson.
5. Find the **Presentation slides** for the lesson and **download** for easier viewing.
6. Click the **Enter virtual classroom** link to join ConferZoom.
7. Log into Opus-II with Putty or ssh command.

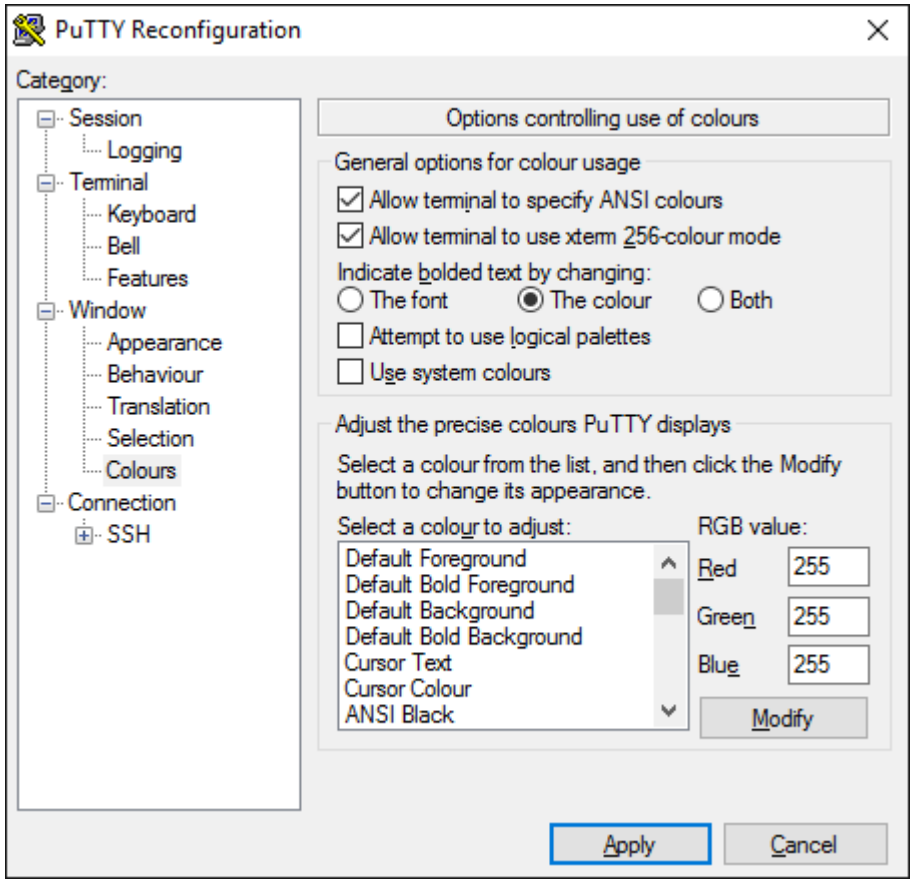
Student checklist - Before class starts

- ❑ *Google*
- ❑ *ConferZoom*
- ❑ *Downloaded PDF of Lesson Slides. I like Foxit Reader so I can take notes using annotations.*



- ❑ *CIS 90 website Calendar page*
- ❑ *One or more login sessions to Opus-II*

Rich's checklist - Putty Colors

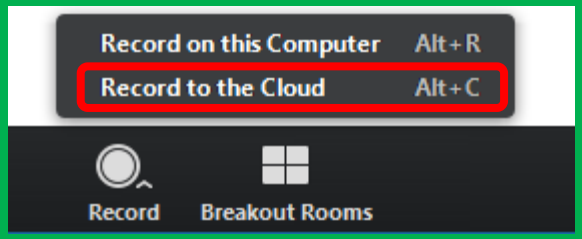


http://looselytyped.blogspot.com/2013/02/zenburn-pleasant-color-scheme-for-putty.html

Putty Colors

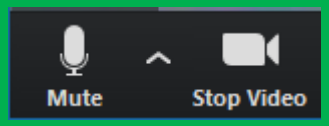
Default Foreground 255 255 255
Default Bold Foreground 255 255 255
Default Background 51 51 51
Default Bold Background 255 2 85
Cursor Text 0 0 0
Cursor Color 0 255 0
ANSI Black 77 77 77
ANSI Black Bold 85 85 85
ANSI Red 187 0 0
ANSI Red Bold 255 85 85
ANSI Green 152 251 152
ANSI Green Bold 85 255 85
ANSI Yellow 240 230 140
ANSI Yellow Bold 255 255 85
ANSI Blue 205 133 63
ANSI Blue Bold 135 206 235
ANSI Magenta 255 222 173
ANSI Magenta Bold 255 85 255
ANSI Cyan 255 160 160
ANSI Cyan Bold 255 215 0
ANSI White 245 222 179
ANSI White Bold 255 255 255

Start



Start Recording

Audio Check



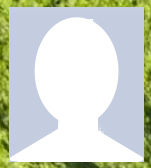
Start Recording

Audio & video Check

CIS 90 - Lesson 9

Instructor: **Rich Simms**
Dial-in: **669-900-6833 (toll)**
Meeting ID: **426 283 384**


Nick


Ryan


Erik


Matt

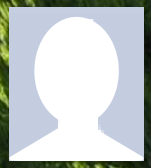

David


Jon

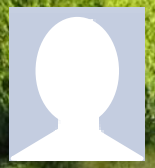

Cheryl

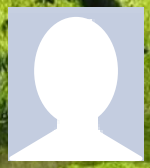

Wais


Tanisha


Daniel


Ohunayo

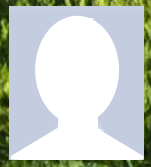

Sequoia


Scott


Lucky


Cole


Shane

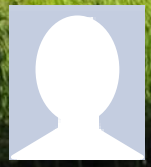

Jim


Joseph


Mark

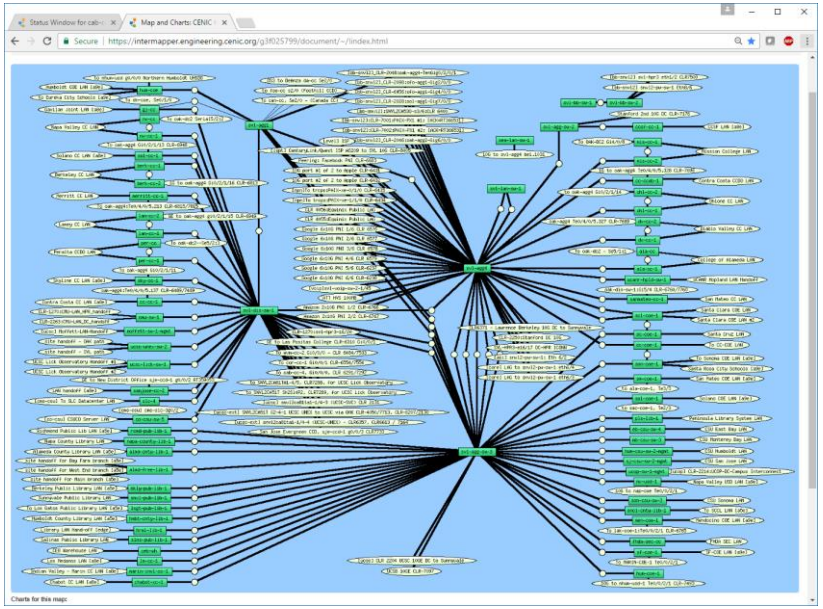

Adina


Evie


Cody

Email me (risimms@cabrillo.edu) a relatively current photo of your face for 3 points extra credit

Network Check



https://intermapper.engineering.cenic.org/g3f025799/document/~/!index.html

First Minute Quiz

Please answer these questions **in the order** shown:

Use ConferZoom White Board

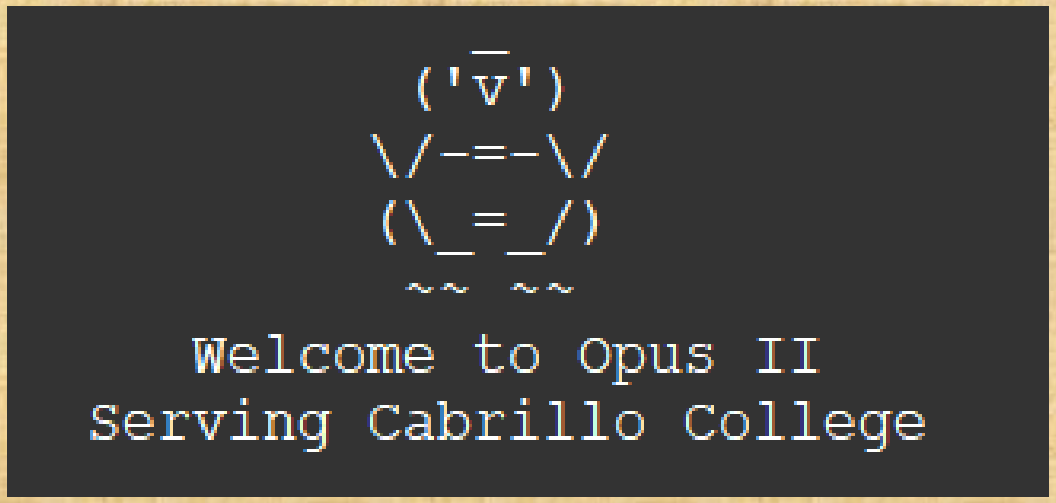
email answers to: risimms@cabrillo.edu

(answers must be emailed within the first few minutes of class for credit)

Review

Objectives	Agenda
• Get ready for the next test • Practice skills • Introduction to processes	• Quiz • Questions • Housekeeping • Linux at school • Linux at home • More on I/O • All together now • Subtle differences • Errors • 2>&1 • More on I/O - programming • umask • More pipeline practice • Pipeline and redirection practice • More on pipelines • Eggs, treats and tricks • Review • Make teams • Flashcard practice • Assignment • Wrap up

Class Activity



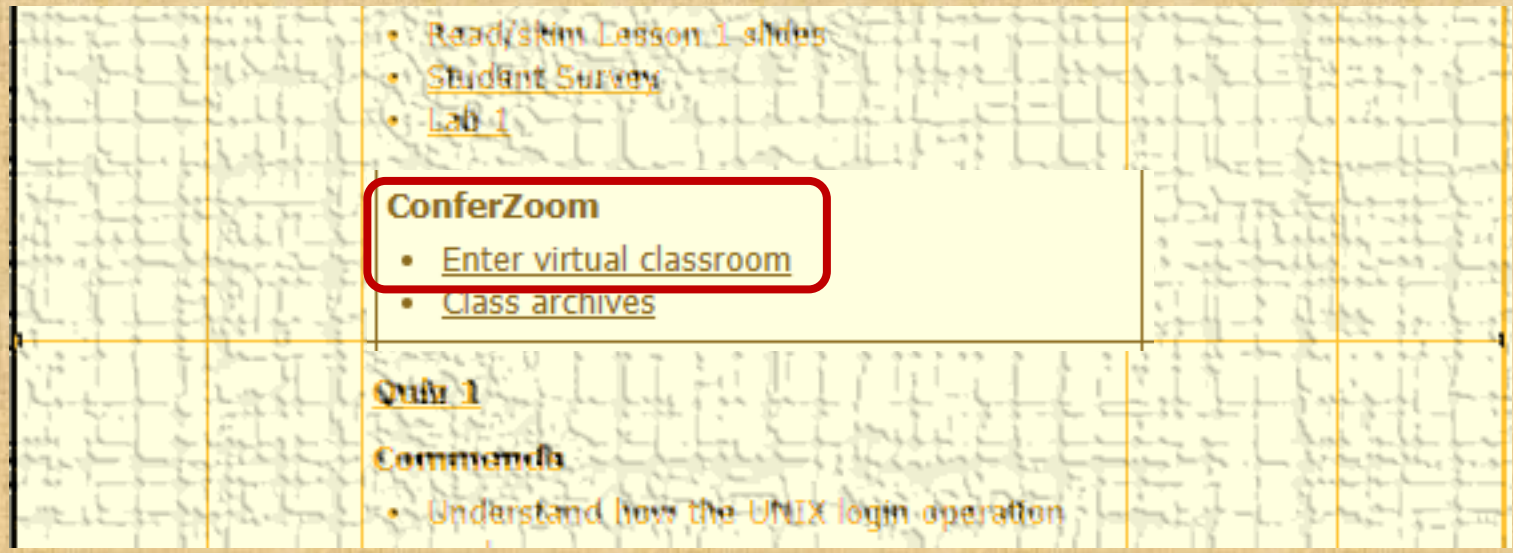
If you haven't already, log into Opus-II

Class Activity

https://simms-teach.com/cis90calendar.php

If you haven't already, download the lesson slides

Class Activity



https://simms-teach.com/cis90calendar.php

If you haven't already, join ConferZoom classroom

Questions

Questions?

Lesson material?

Labs? Tests?

How this course works?

- Graded work & tests in home directories
- Answers in /home/cis90/answers

Who questions much, shall learn much, and retain much.
- Francis Bacon

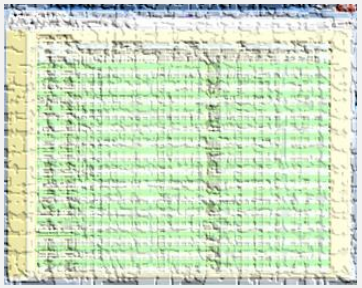
If you don't ask, you don't get.
- Mahatma Gandhi

Chinese Proverb	他問一個問題，五分鐘是個傻子，他不問一個問題仍然是一個傻瓜永遠。 *He who asks a question is a fool for five minutes; he who does not ask a question remains a fool forever.*

Review your progress in the course

Check the website Grades page

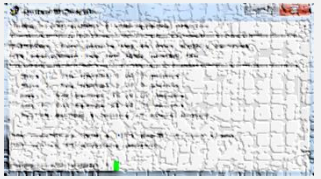
http://simms-teach.com/cis90grades.php



Or check on Opus-II

checkgrades *codename*

(where codename is your LOR codename)



Written by Jesse Warren a past CIS 90 Alumnus

- **Send me your survey to get your LOR codename.**
- **Graded labs and tests are in your home directories.**

Percentage	Total Points	Letter Grade	Pass/No Pass
90% or higher	504 or higher	A	Pass
80% to 89.9%	448 to 503	B	Pass
70% to 79.9%	392 to 447	C	Pass
60% to 69.9%	336 to 391	D	No pass
0% to 59.9%	0 to 335	F	No pass

At the end of the term I'll add up all your points and assign you a grade using this table

Points that could have been earned:	
6 quizzes:	18 points
6 labs:	180 points
1 test:	30 points
2 forum quarters:	40 points
Total:	**268 points**

Extra Credit

On the forum

Be sure to monitor the forum as I may post extra credit opportunities without any other notice!

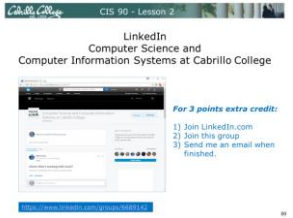
On some labs

Extra credit (2 points)

For a small taste of what you would learn in CIS 191 let's add a new user to your Arya VM. Once added we will see how the new account is represented in */etc/passwd* and */etc/shadow*.

1. Log into your Arya VM as the cis90 user. Make sure it's your VM and not someone else's.
2. Install the latest updates:
   ```
   sudo apt-get update
   sudo apt-get upgrade
   ```
3. Add a new user account for yourself. You may make whatever username you wish. The example below shows how Benji would make the same username he uses on Opus:
   ```
   sudo useradd -G sudo -c "Benji Simms" -m -s /bin/bash simben90
   ```

In lesson slides (search for extra credit)

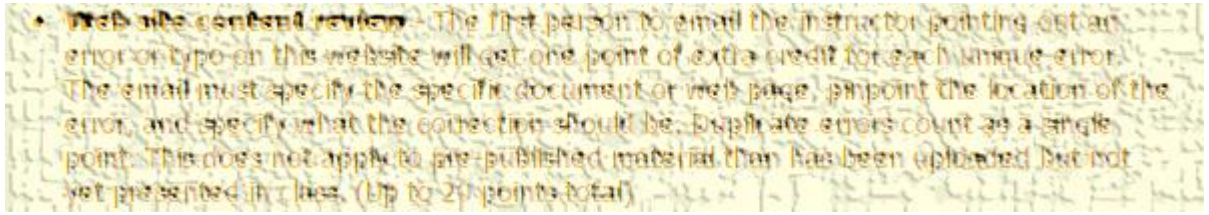


On the website

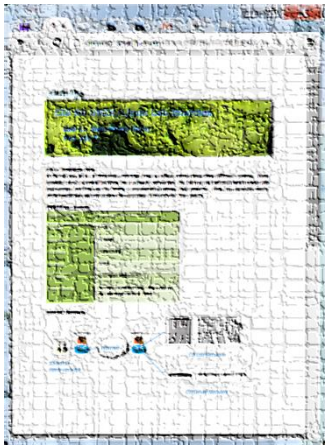
http://simms-teach.com/cis90grades.php

For some flexibility, personal preferences or family emergencies there is an additional 90 points available of **extra credit** activities.

http://simms-teach.com/cis90extracredit.php



Lab Assignments -- Pearls of Wisdom



- Don't wait till the last minute to start.
- Plan for things to go wrong and give yourself time to ask questions and get answers.
- The *slower* you go the *sooner* you will be finished.
- A few minutes reading the forum can save you hour(s).
- Line up materials, references, equipment and software ahead of time.
- It's best if you fully understand each step as you do it. Use Google or refer back to lesson slides to understand the commands you are using.
- Keep a growing cheat sheet of commands and examples.
- Study groups are very productive and beneficial.
- Use the forum to collaborate, ask questions, get clarifications and share tips you learned while doing a lab.
- **Late work is not accepted** so submit what you have for partial credit.

Getting Help When Stuck on an Assignment

- Google the topic/error message.
- Search the Lesson Slides (they are PDFs) for a relevant example on how to do something.
- Check the forum. Someone else may have run into the same issue and found a way past it. If not start a new topic, explain what you are trying to do and what you have tried so far.
- Talk to a tutor/assistant at the CTC (room 1403) or CIS Lab (STEM Center).
- Come see me during my office or lab hours:
  https://www.cabrillo.edu/salsa/listing.php?staffId=1426
  I'm in the CTC (room 1403) every Tuesday from 3:30-6:00 pm.
- Make use of the Open Questions time at the start of every class.
- Make a cheat sheet of commands and examples so you never again get stuck on the same thing!

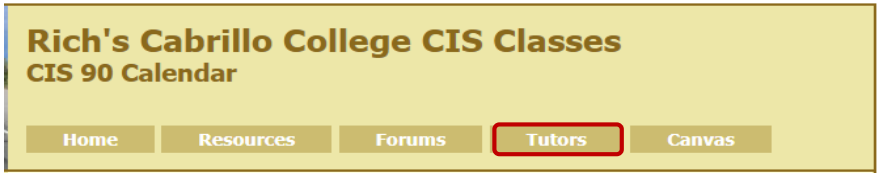
CIS Labs always involve some troubleshooting!

Help Available!

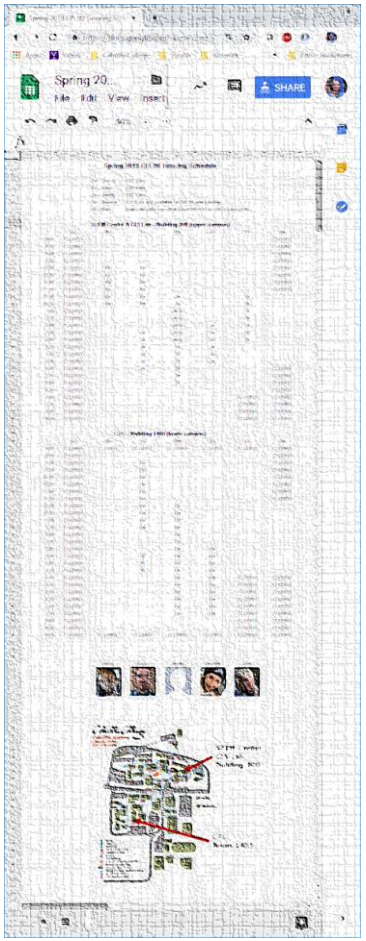
In the CTC and CIS Lab

CIS Lab
in STEM Center
Building 800

CTC
Room 1403



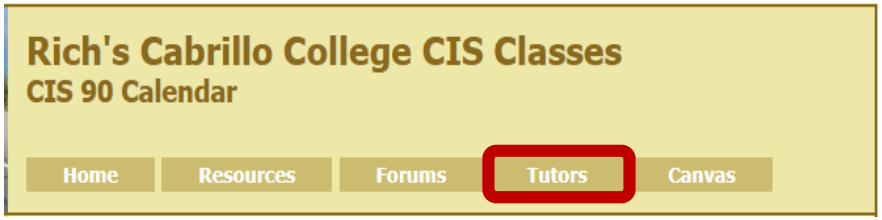
To see tutor schedule, click the Tutors link on the website.



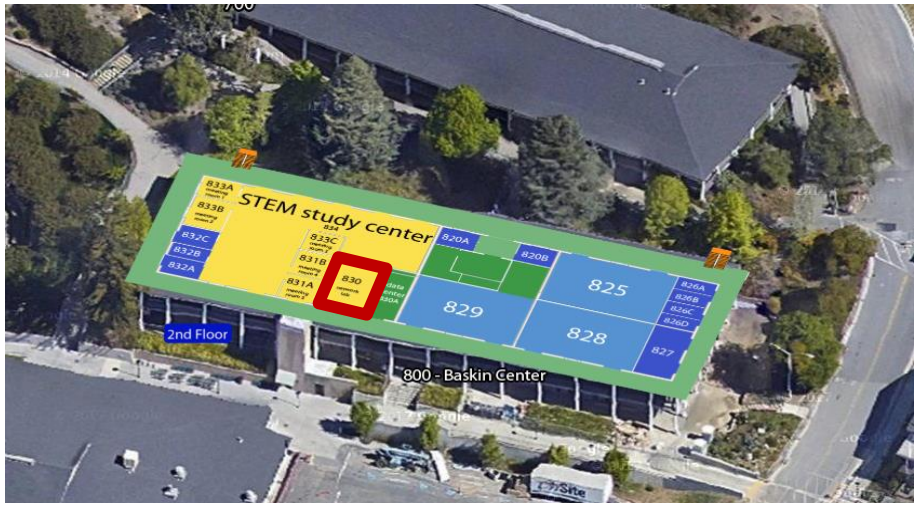
Instructors, tutors and equipment are available for CIS students to work on assignments.

Help Available!

In the CTC and CIS Lab



To see tutor schedule, click the Tutors link on the website.



The CIS Lab is in the STEM center (Building 800)

Room 1403 is in the CTC (Building 1400)

The slippery slope

1) If you didn't submit the last lab ...

2) If you were in class and didn't submit the last quiz ...

3) If you didn't send me the student survey assigned in Lesson 1 ...

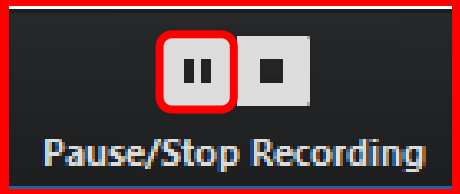
4) If you haven't made a forum post in the last quarter of the course ...

5) If you had trouble doing the last test ...

Please contact me by email, see me during my office hours or when I'm in the CTC

Email: risimms@cabrillo.edu

Housekeeping



Pause Recording

Audio Check

Roll Call

If you are watching the archived video please email me to let me know you were here.

risimms@cabrillo.edu

Overlap Students

Don't forget to update the Google Docs Log when watching the recording

Resume Recording

Audio Check

Housekeeping

1. Lab 7 due 11:59PM tonight -- **don't forget to submit your latest version!**

2. Read your Opus-II email for Lab 7 submission status.

3. A **check7** script is available.

4. Fine Print:
   Test #2 is scheduled for our next class!

Test #2 will happen during our next class!

Practice test available after class.

Test #2 is scheduled for our next next class!

Practice test available after class.


Test #2 is scheduled for our next class!
Practice test available after class.

Test #2 next week

30 points, plus some extra credit:

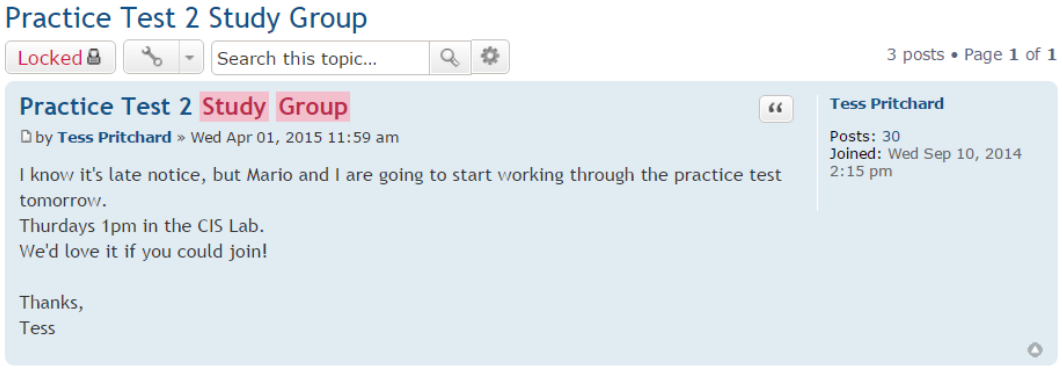
- Open book, open notes, open computer.
- You must work alone and not help or receive help from others.
- Online timed 60 minute test using Canvas.
- To be taken during the last hour of class.
- Online "archive watching" students that work can take it later in the day but it must be completed by 11:59 PM.
- Practice test is available after class on Canvas.

Note: Practice test systems shutdown before real test starts!

How to pass Test #2 with flying colors

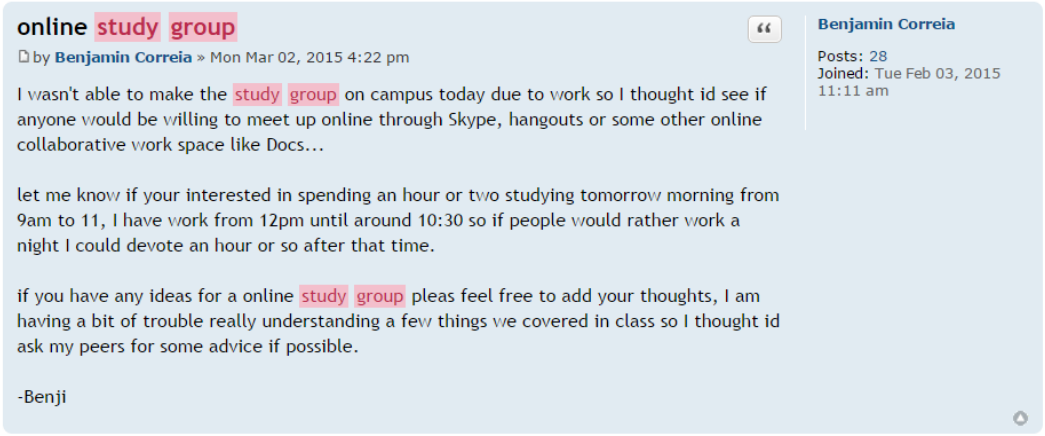
- Keep taking the practice test until you can complete each question in 30 seconds or less. Use the **./restore** command to restore your directory on the test system to the original state.
- On each pass taking the practice test update your crib sheet so your have clear and accurate notes on how to do each task.
- Use the forum to discuss your approaches and results with classmates.
- If a question takes longer than 30 seconds ask for help. You can ask for help on the forum, see me during office hours, work with a tutor in the STEM center, join a study group or all of the above!
- Use the flashcards on the course website and rework any labs and previous tests you want to better understand.
- Tip: Use the **-v** option on **chmod**, **mv**, **cp** and **rm** commands to see what actually happened.
- But most important … DON'T WAIT TILL THE LAST MINUTE to prepare!

Use the forum to arrange study groups



Example Fall 2014 post to meet in the STEM center to study for a CIS 90 test

Example Fall 2014 post to meet online to study for a CIS 90 test



To get notifications of new forum posts

Subscribe to the forum to get email notifications of new posts

After logging in:

1. Go to the CIS 90 class forum.
2. At the bottom of the page, click the "Subscribe forum" link on the lower left. When subscribed you get email notifications when new posts are made.
3. To unsubscribe, click it again.

Unsubscribed looks like this.

Subscribed looks like this.

Heads up on Final Exam

Test #3 (final exam) is **Wednesday May 22, 7-9:50AM**

Wed	5/22	**Test #3 (the final exam)** **Time** • WEDNESDAY 7:00AM - 9:50AM in Room 828 or online **Materials** • Presentation slides (download) • Test (canvas) **ConferZoom** • Enter virtual classroom • Class archives		5 posts Lab X1 Lab X2

Extra credit labs and final posts due by 11:59PM

- All students will take the test at the same time. The test starts at **7:00AM** must be completed by **9:50AM**.
- Working and long distance students can take the test online via ConferZoom and Canvas.
- Working students will need to plan ahead to arrange time off from work for the test.
- Test #3 is **mandatory** (even if you have all the points you want)

SPRING 2019 FINAL EXAMINATIONS SCHEDULE
MAY 20 TO MAY 25

DAYTIME FINAL SCHEDULE

Daytime Classes: All times in bold refer to the beginning times of classes. **MW/Daily** means Monday alone, Wednesday alone, Monday and Wednesday **or any 3** or more days in any combination. **TTH** means Tuesday alone, Thursday alone, or Tuesday and Thursday. **Classes meeting other combinations of days and/or hours not listed must have a final schedule approved by the Division Dean.**

STARTING CLASS TIME / DAY(S)	EXAM HOUR	EXAM DATE
Classes starting between:		
6:30 am and 8:55 am, MW/Daily	7:00 am-9:50 am	Monday, May 20
9:00 am and 10:15 am, MW/Daily	7:00 am-9:50 am	Wednesday, May 22

CIS 90 Introduction to UNIX/Linux

Provides a technical overview of the UNIX/Linux operating system, including hands-on experience with commands, files, and tools.
Recommended Preparation: CIS 1L or CIS 72.
Transfer Credit: Transfers to CSU;UC

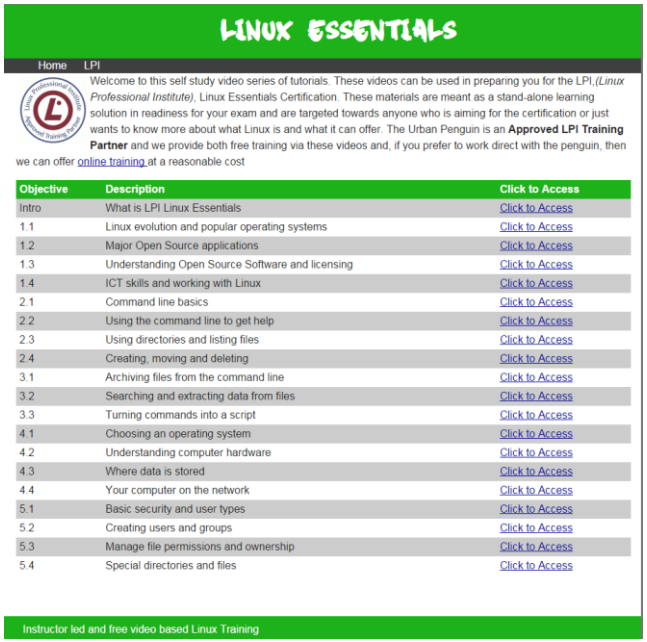
Section	Days	Times	Units	Instructor	Room
1	W	9:00AM-12:05PM	3.00	R.Simms	OL
Section 1 is an ONLINE course. Meets weekly throughout the semester online during the scheduled times by remote technology with an additional 50 min arranged online lab per week. For details, see instructor's web page at go.cabrillo.edu/online.					
2	W	9:00AM-12:05PM	3.00	R.Simms	828
&	Arr.	Arr.		R.Simms	OL
Section 2 is a Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 50 min online lab per week. For details, see instructor's web page at go.cabrillo.edu/online.					

LPI Linux Essentials Certificate

Linux Essentials Certificate of Achievement

Objective	# of Questions	Cabrillo	Urban Penguin	NDG Linux Essentials
Topic 1: The Linux Community and a Career in Open Source				
1.1 Linux Evolution and Popular Operating Systems	2	CIS90 Lesson 1	1.1	Module 1
1.2 Major Open Source Applications	2	CIS90 Lesson 1	1.2	Module 2
1.3 Understanding Open Source Software and Licensing	1	CIS90 Lesson 1	1.3	Module 2
1.4 ICT Skills and Working in Linux	2	not covered	1.4	Module 3
Topic 2: Finding Your Way on a Linux System				
2.1 Command Line Basics	2	CIS90 Lesson 2	2.1	Module 4
2.2 Using the Command Line to Get Help	2	CIS90 Lesson 2	2.2	Module 5
2.3 Using Directories and Listing Files	2	CIS 90 Lesson 4	2.3	Module 6
2.4 Creating, Moving and Deleting Files	2	CIS90 Lesson 5	2.4	Module 6
Topic 3: The Power of the Command Line				
3.1 Archiving Files on the Command Line	2	CIS 90 Lesson 14	3.1	Module 7
3.2 Searching and Extracting Data from Files	4	CIS 90 Lesson 8	3.2	Module 8
3.3 Turning Commands into a Script	4	CIS 90 Lesson 13 & 14	3.3	Module 9
Topic 4: The Linux Operating System				
4.1 Choosing an Operating System	1	not covered	4.1	Module 1
4.2 Understanding Computer Hardware	2	CIS 90 Lesson 1	4.2	Module 10
4.3 Where Data is Stored	3	CIS 90 Lesson 1	4.3	Module 11
4.4 Your Computer on the Network	2	CIS 192	4.4	Module 12
Topic 5: Security and File Permissions				
5.1 Basic Security and Identifying User Types	2	CIS 191	5.1	Module 13
5.2 Creating Users and Groups	2	CIS 191	5.2	Module 14
5.3 Managing File Permissions and Ownership	2	CIS 90 Lesson 7	5.3	Module 15
5.4 Special Directories and Files	1	CIS 90 Lesson 4	5.4	Module 16

The Urban Penguin



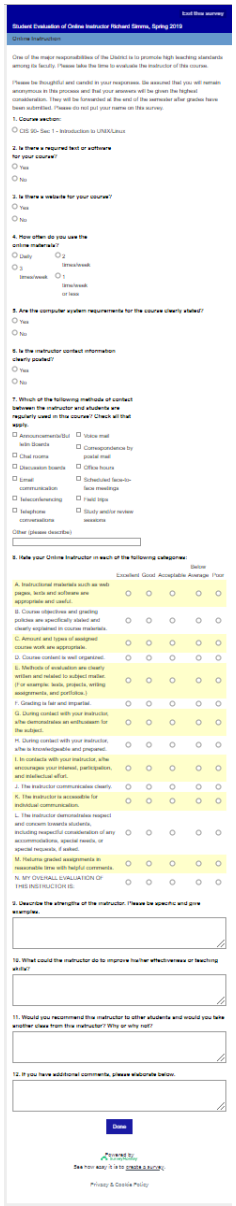
http://www.theurbanpenguin.com/lpi/le.html

No registration, no logging in, just click and watch the videos

NDG Linux Essentials via Cisco Networking Academy

https://www.netacad.com/

Complete course with reading, live VM and tests. Contact me if you would like a student account for the NDG Linux Essentials course.


Exit this survey

Student Evaluation of Online Instructor Richard Simms, Spring 2019

Online Instruction

One of the major responsibilities of the District is to promote high teaching standards among its faculty. Please take the time to evaluate the instructor of this course.

Please be thoughtful and candid in your responses. Be assured that you will remain anonymous in this process and that your answers will be given the highest consideration. They will be forwarded at the end of the semester after grades have been submitted. Please do not put your name on this survey.

1. Course section:
CIS 90- Sec 1 - Introduction to UNIX/Linux

2. Is there a required text or software for your course?
Yes
No

3. Is there a website for your course?
Yes
No

4. How often do you use the online materials?
Daily
2 times/week
3 times/week
1 time/week or less

5. Are the computer system requirements for the course clearly stated?
Yes
No

6. Is the instructor contact information clearly posted?
Yes
No

7. Which of the following methods of contact between the instructor and students are regularly used in this course? Check all that apply.
Announcements/Bulletin Boards
Chat rooms
Discussion boards
Email communication
Telecontferencing
Telephone conversations
Voice mail
Correspondence by postal mail
Office hours
Scheduled face-to-face meetings
Field trips
Study and/or review sessions
Other (please describe)

8. Rate your Online Instructor on each of the following categories:

	Excellent	Good	Acceptable	Average	Below Average / Poor
A. Instructional materials such as web pages, texts and software are appropriate and useful.					
B. Course objectives and grading policies are specifically stated and clearly explained in course materials.					
C. Amount and types of assigned course work are appropriate.					
D. Course content is well organized.					
E. Methods of evaluation are clearly written and related to subject matter. (For example: tests, projects, writing assignments, and portfolios.)					
F. Grading is fair and impartial.					
G. During contact with your instructor, s/he demonstrates an enthusiasm for the subject.					
H. During contact with your instructor, s/he is knowledgeable and prepared.					
I. In contacts with your instructor, s/he encourages your interest, participation, and intellectual effort.					
J. The instructor communicates clearly.					
K. The instructor is accessible for individual communication.					
L. The instructor demonstrates respect and concern towards students, including respectful consideration of any accommodations, special needs, or special requests, if asked.					
M. Returns graded assignments in reasonable time with helpful comments.					
N. MY OVERALL EVALUATION OF THIS INSTRUCTOR IS:					

9. Describe the strengths of the instructor. Please be specific and give examples.

10. What could the instructor do to improve his/her effectiveness or teaching skills?

11. Would you recommend this instructor to other students and would you take another class from this instructor? Why or why not?

12. If you have additional comments, please elaborate below.

Done

Powered by SurveyMonkey
See how easy it is to create a survey.
Privacy & Cookie Policy

Your turn to grade me!

1. March 18th - NAS Office sends online SurveyMonkey survey to all CIS 90 students. Please complete it by April 17th.

2. I've added the survey link to the website Calendar page as well (see Lesson 9).

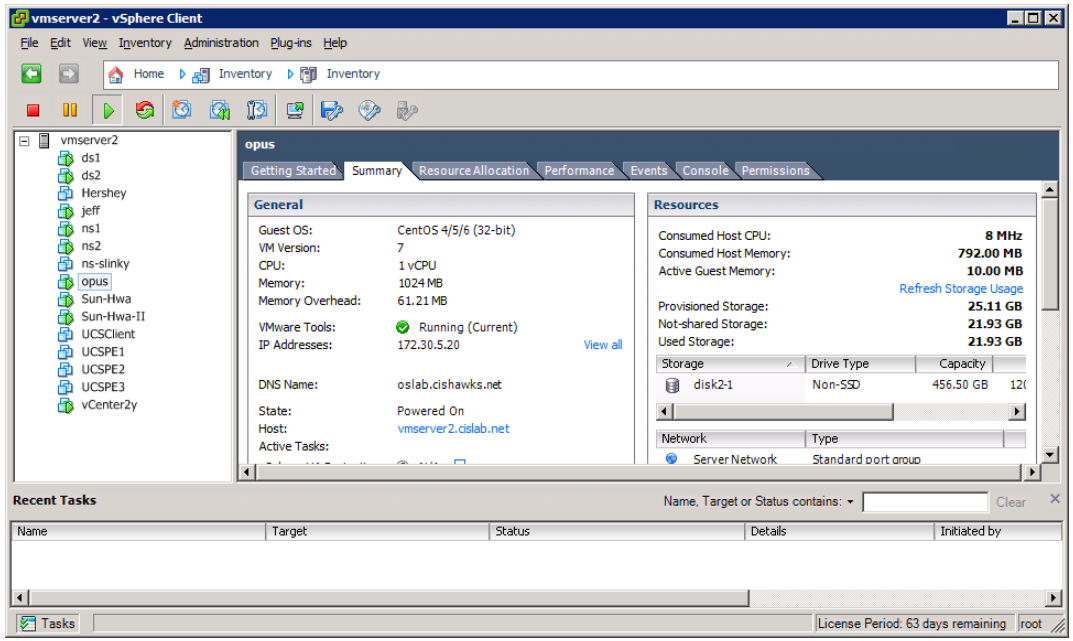
3. April 3rd - Department Chair (Mike) will visit our class to observe.

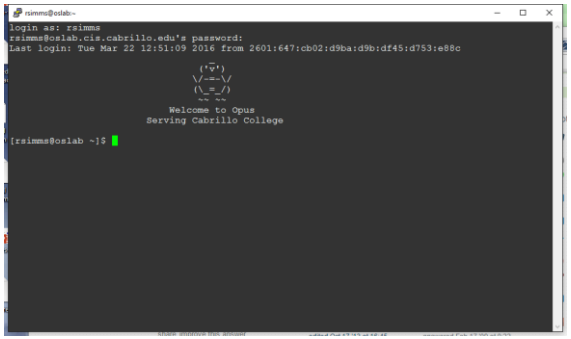
Linux at School

Our Opus-II server on campus

Dell R610 Server

VMware vSphere Client





Opus-II is a VM running on one of the Vmware ESXi servers in the CIS Datacenter

Linux at Home

USB "Live" Linux Boot USB Drive

Allows you to use or try out Linux on an existing computer without installing it

1) 

Get the Linux distros of your choice
Or see: http://iso.linuxquestions.org/

2) 

Get a USB flash drive

3) Google "boot live linux from usb" for instructions
Or see: http://www.pendrivelinux.com/yumi-multiboot-usb-creator/

4) 

Configure your BIOS to boot from USB then select the Operating System as your computer boots up

USB "Live" Linux Multi-Boot USB Flash Drive

Windows

CentOS

Kali

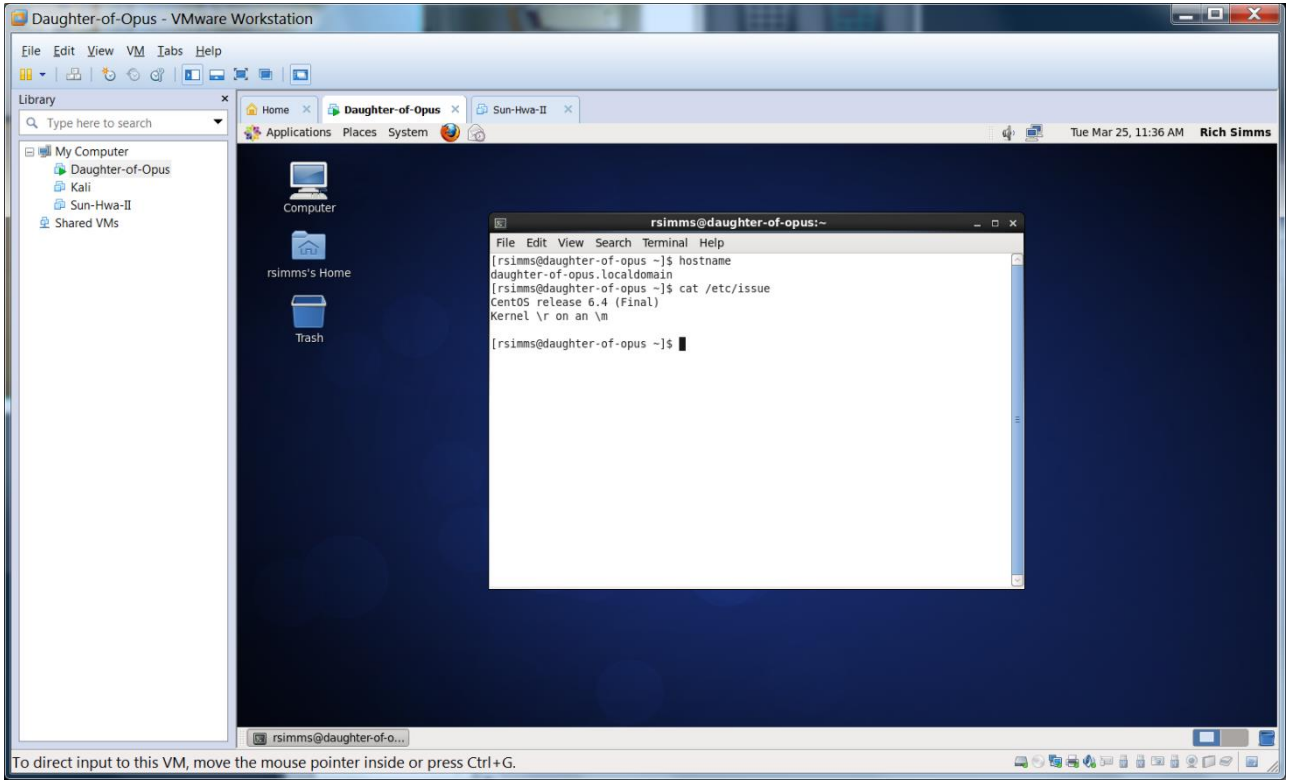
YUMI formatted Flash Drive
(www.pendrivelinux.com)

Linux Mint

Ubuntu

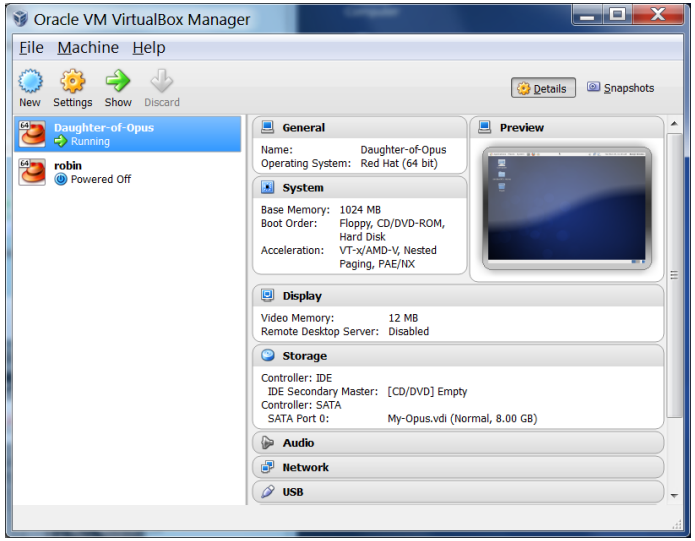
Allows you to use or try out Linux on an existing computer without installing it

VMware Workstation (PC) or Fusion (Mac)

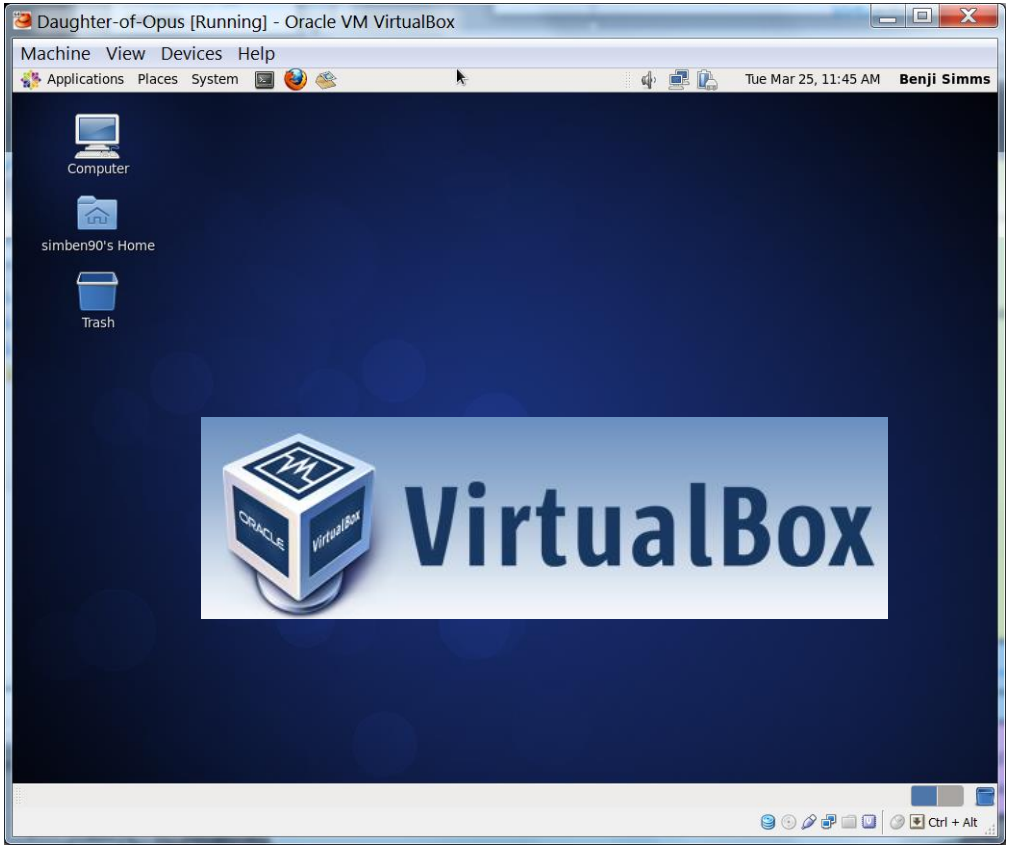


One Daughter-of-Opus is a VM running on my laptop using VMware Workstation (expires in one year)

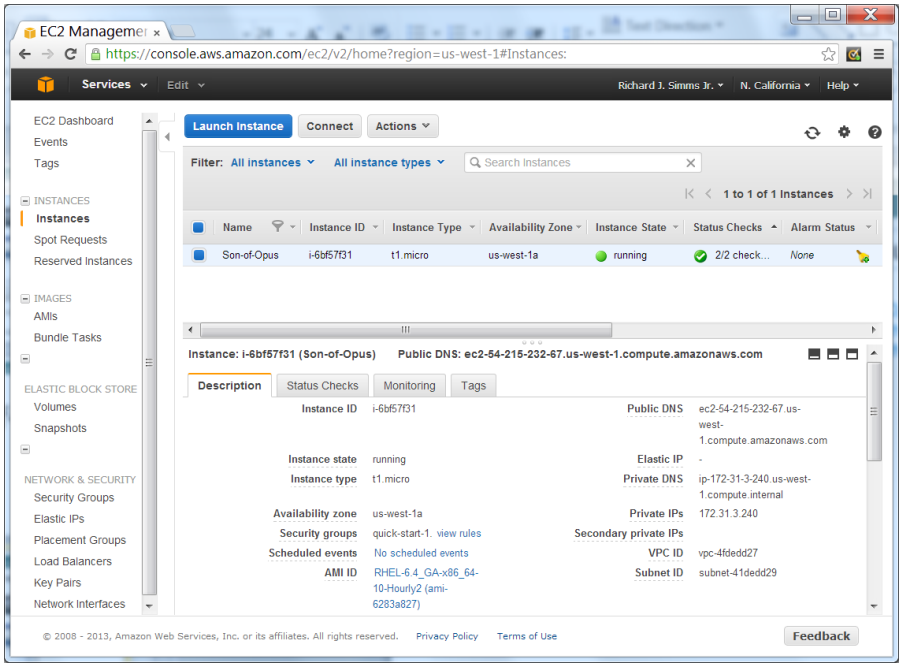
VirtualBox



This Daughter-of-Opus is a VM running on my laptop using Oracle VirtualBox (never expires)



Amazon Web Services

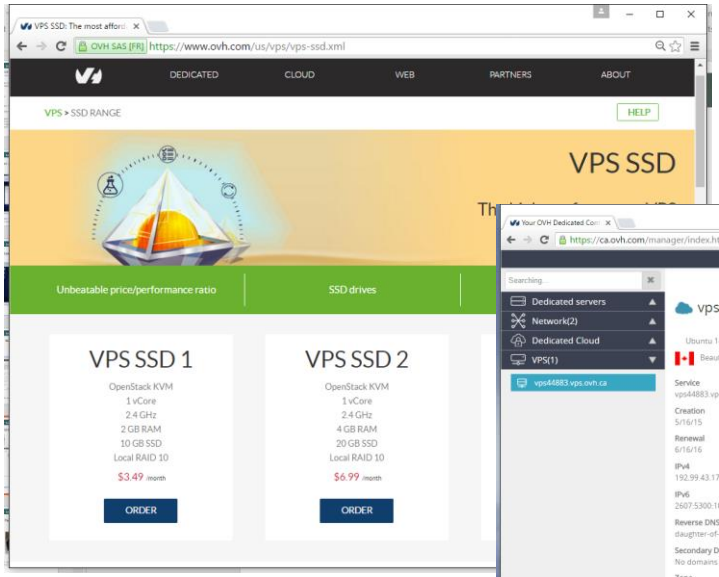


Son-of-Opus is a VM running on Amazon Web Services

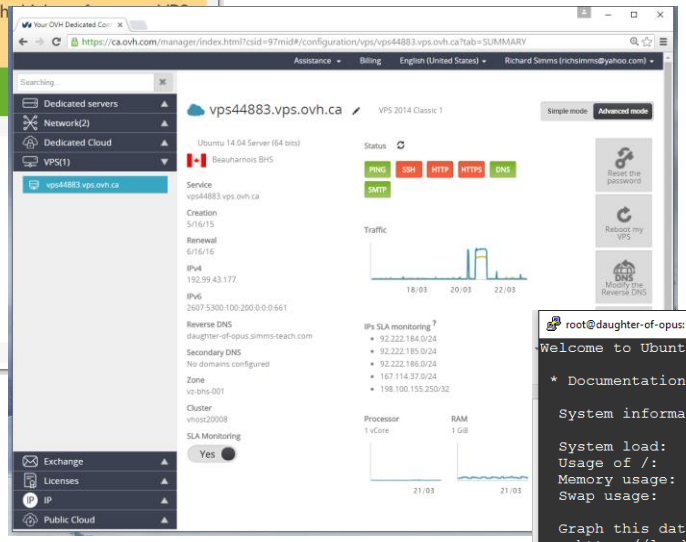
Single EC2 "instance" is free for a year, then about $60 per month after that.

OVH.com

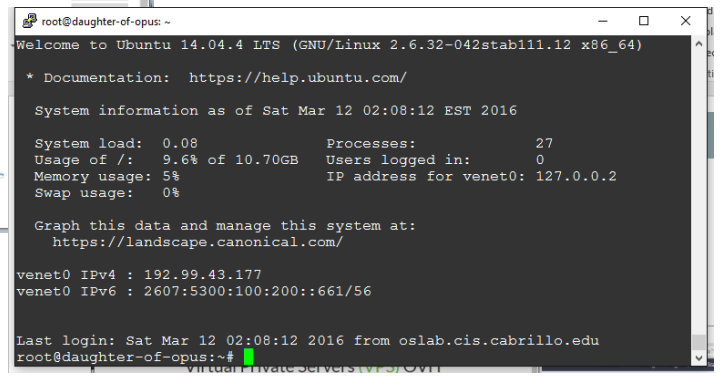
Purchase



Configure



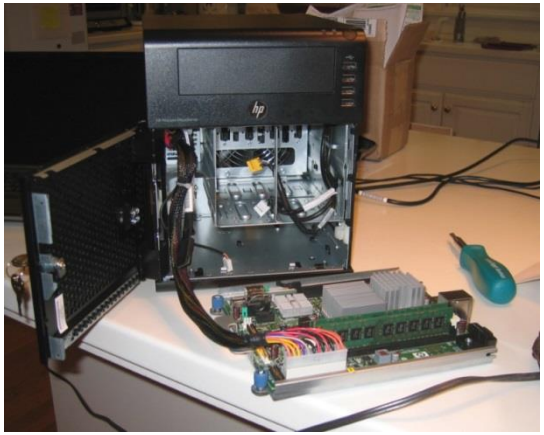
Use



Virtual private servers like daughter-of-opus used on Test #1 costs $3 per month

Small Form Factor Servers

HP Microserver



https://www.hpe.com/us/en/product-catalog/servers/proliant-servers.filters-facet_subbrand_url:ProLiant-MicroServer.hits-12.html

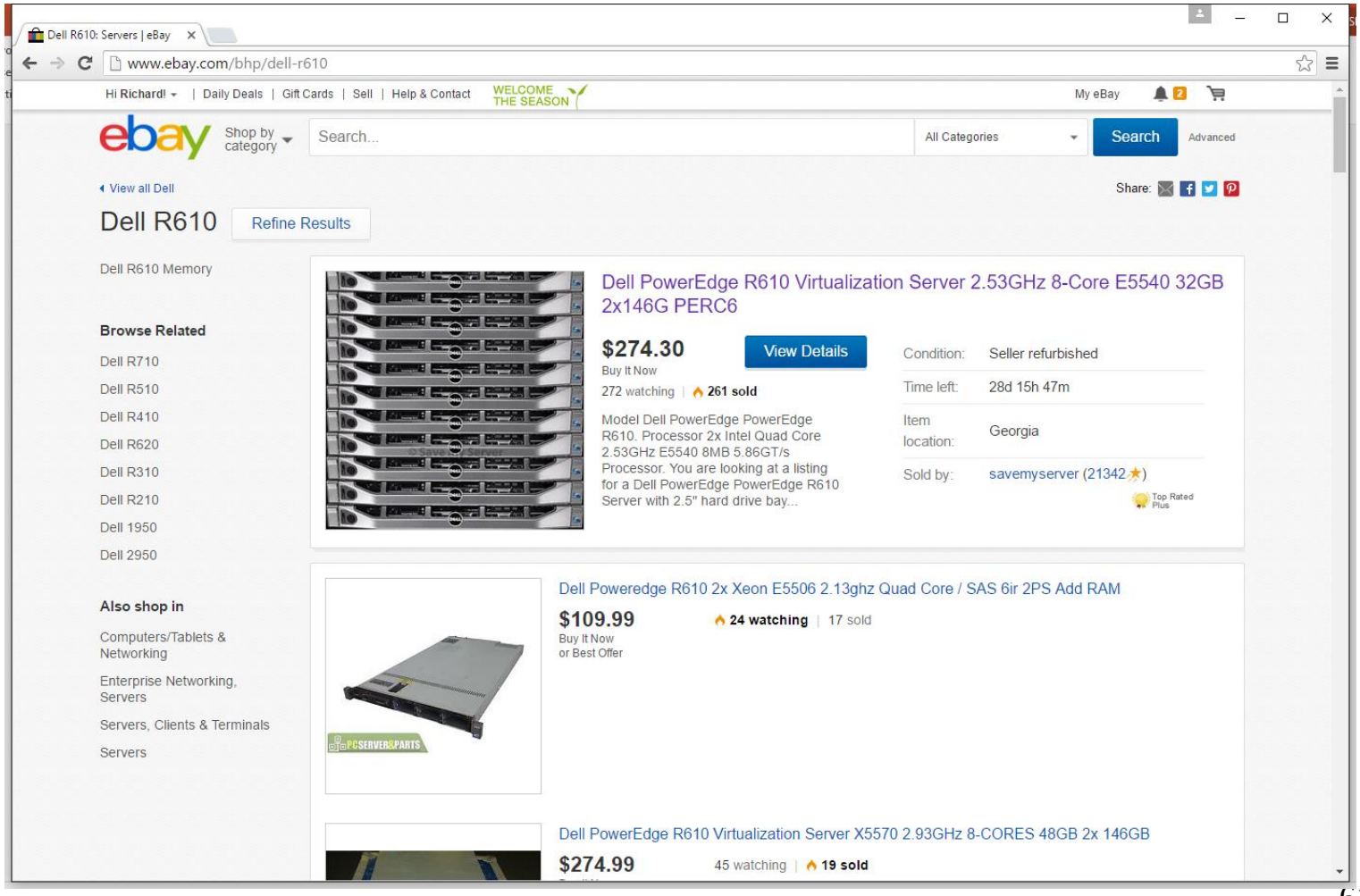
vmware ESXi

SuperMicro

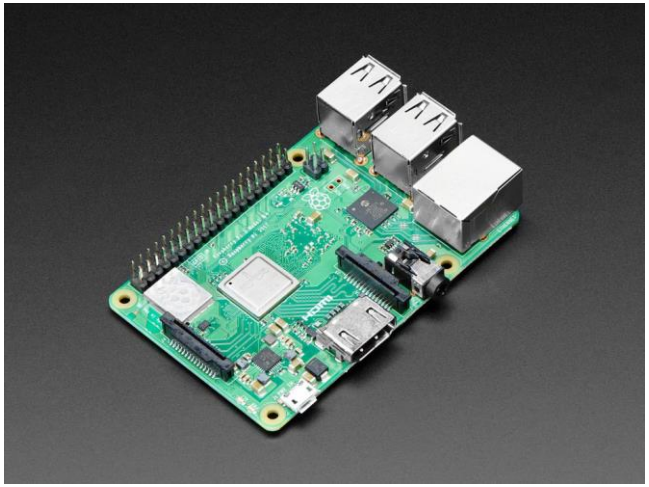
https://tinkertry.com/my-tinkertry-d-xeon-d-bundle-2-supermicro-superserver-bundle-2-of-joy

Comparatively inexpensive "bare bones" servers that come without memory, hard drives or an operating system

Fantastic Bargains on EBay

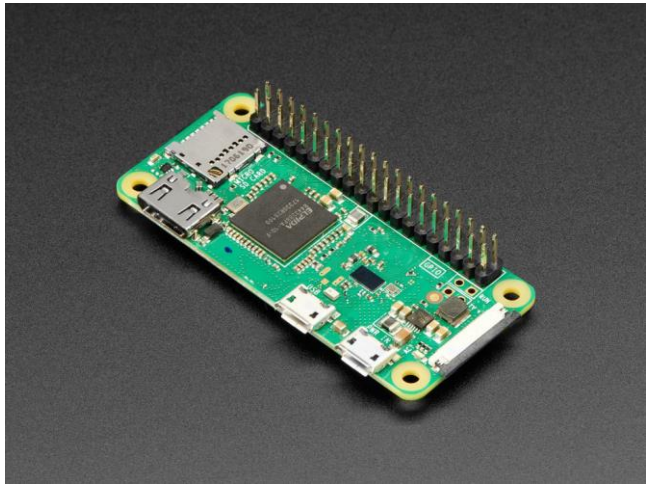


Raspberry Pi



Raspberry Pi 3 Model B+

https://www.adafruit.com/product/3775

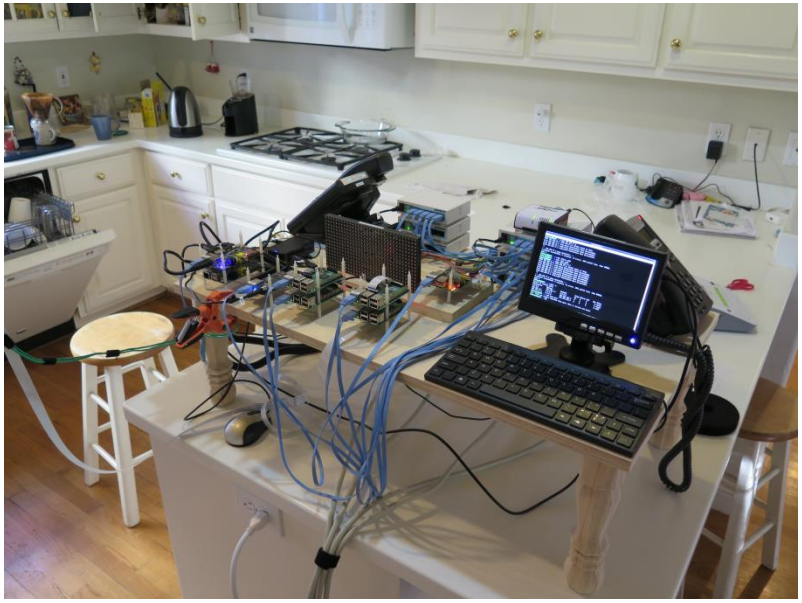


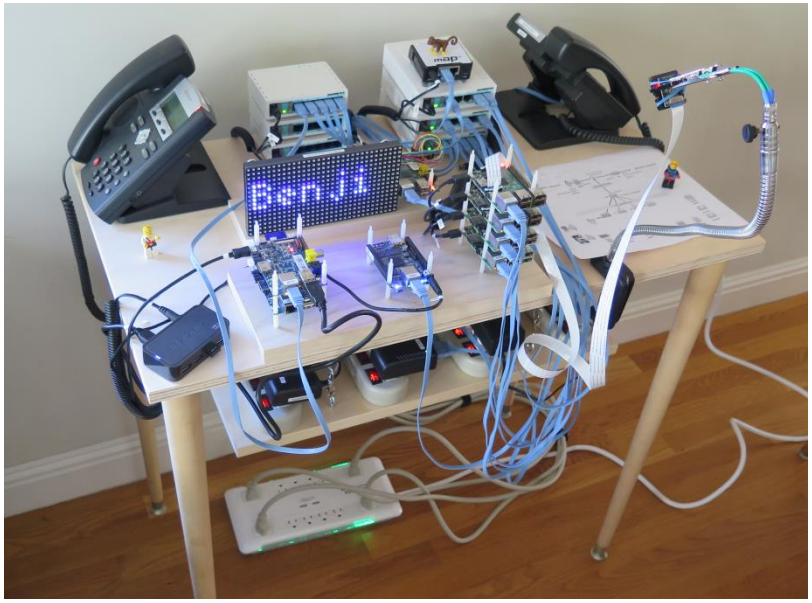
Raspberry Pi Zero WH

https://www.adafruit.com/product/3708

Micro Datacenters

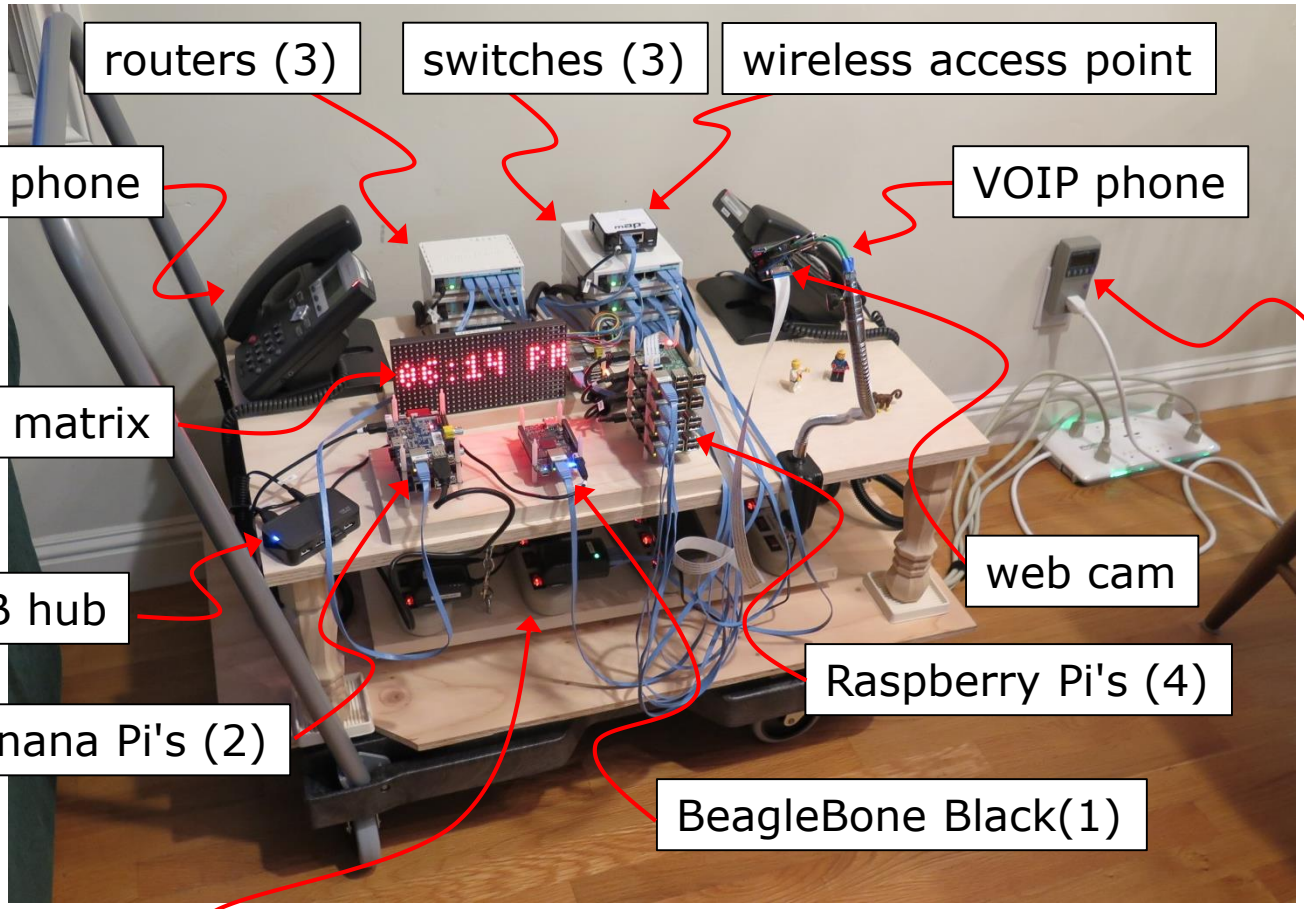
A very tiny home made datacenter





Low cost way to practice all your CIS network and system skills

A very tiny home made datacenter

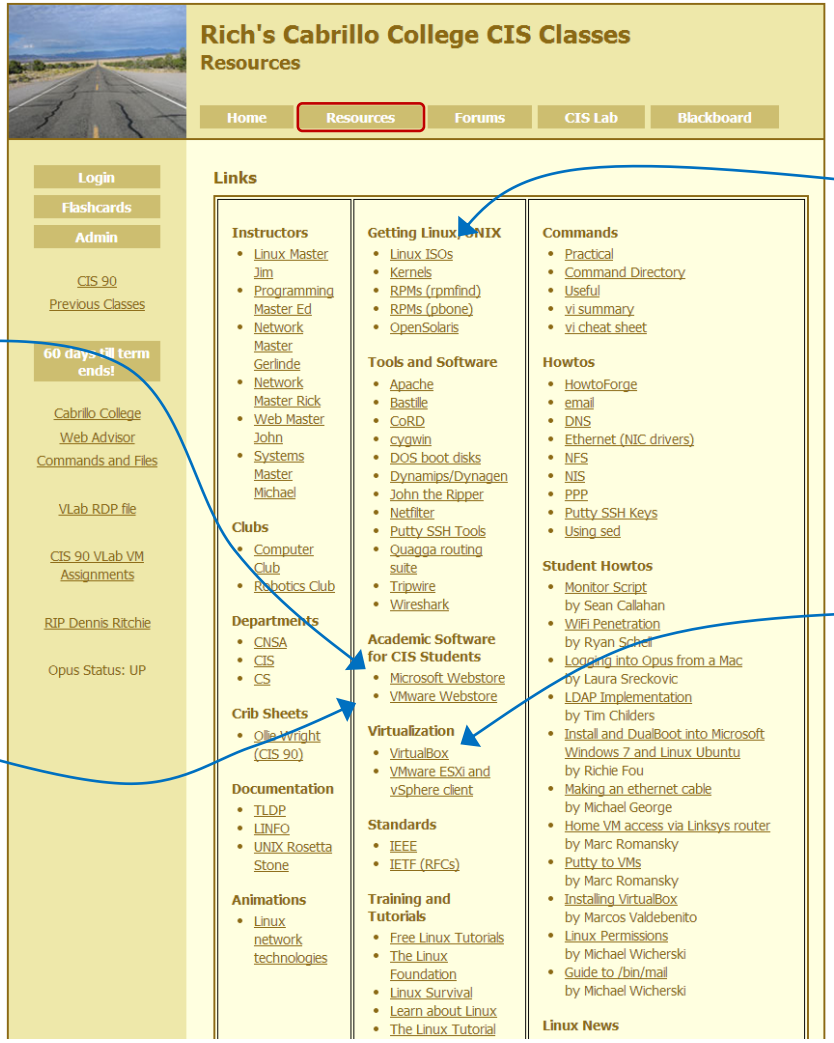


39 watts total

See "Your Name in Lights" post in forum

Microsoft Software (Academic)

http://simms-teach.com/resources.php



Linux Distros (ISOs)

VirtualBox

VMware Software (Academic)

More on I/O
(input/output)

Input and Output
File Redirection

The 3 standard UNIX file descriptors:

Name	Integer Value
stdin (**st**an**d**ard **in**)	0
stdout (**st**an**d**ard **out**)	1
stderr (**st**an**d**ard **err**or)	2

*Every process is provided with three file descriptors: **stdin**, **stdout** and **stderr***

Input and Output
File Redirection

*The input and output of a program can be **redirected** to and from other files as follows:*

0< *filename*

Redirects **stdin**, input will now come from *filename* rather than the keyboard.

1> *filename*

Redirects **stdout**, output will now go to *filename* instead of the terminal.

2> *filename*

Redirects **stderr**, error messages will now go to *filename* instead of the terminal.

>> *filename*

Redirects **stdout**, output will now be appended to *filename*.

2>> *filename*

Redirects **stderr**, output will now be appended to *filename*.

The redirection is specified on the command line

Shell prints this to prompt user to enter a command

Shell parses this command line

***Redirection** connects **stdin**, **stdout** and **stderr** to non-default devices*

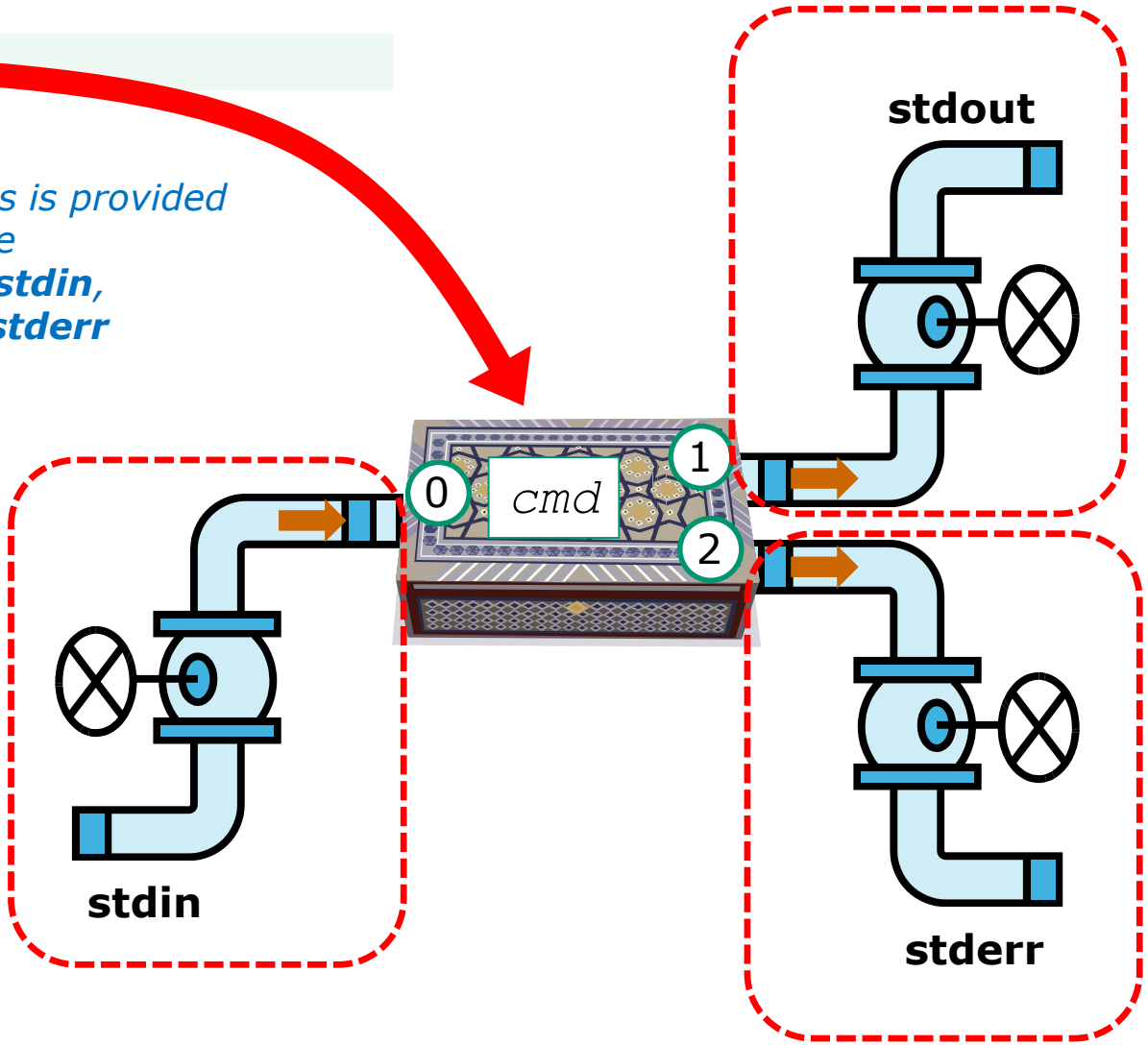
Examples

```
/home/cis90/simben $ cat
/home/cis90/simben $ cat -A letter
/home/cis90/simben $ cat   < letter
/home/cis90/simben $ cat -b < letter > out
/home/cis90/simben $ cat   bogus 2> /dev/null
/home/cis90/simben $ cat -e < bogus 2> /dev/null
/home/cis90/simben $ cat -e < letter > out 2> /dev/null
```

A program loaded into memory becomes a **process**

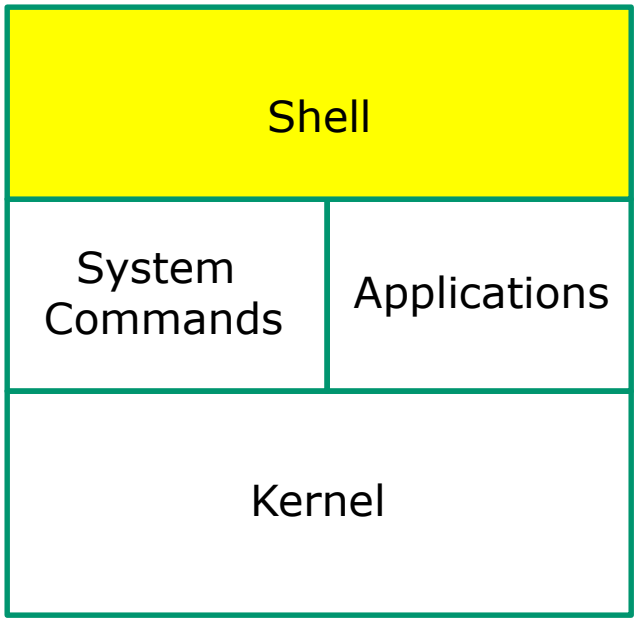
```
$ cmd
```

*Every process is provided with three file descriptors: **stdin**, **stdout** and **stderr***



All Together Now Example

Life of the Shell



1) Prompt
2) Parse
3) Search
4) Execute
5) Nap
6) Repeat

Shell generates the prompt string

1) Prompt
2) Parse
3) Search
4) Execute
5) Nap
6) Repeat

The shell begins by echoing a **prompt** string to your terminal device:

- Your specific terminal device can be identified by using the **tty** command.
- The format of the prompt is defined by the contents of the PS1 variable (show with **echo $PS1**).

```
/home/cis90/simben $
```

The prompt string. In this case the PS1 variable is set to '$PWD $ ' which results in a prompt that shows the current location in the file tree followed by a blank, a $, and another blank.

Activity

1) Prompt
2) Parse
3) Search
4) Execute
5) Nap
6) Repeat

The prompt is defined by your PS1 variable

1. Look at the contents of your PS1 variable: **echo $PS1**
2. Look at the contents of your PWD variable: **echo $PWD**
3. Send me and yourself the contents of your prompt variable:
   echo $PS1 | mail -s "my prompt" rsimms $LOGNAME

Paste the value of your PWD variable into the chat window when finished

User responds to the prompt by entering a command

1) Prompt
2) Parse
3) Search
4) Execute
5) Nap
6) Repeat

Following the prompt, the user then enters a command followed by the Enter key:

- The Enter key generates a <newline> which is a "unprintable" shell metacharacter. All metacharacters have special meanings to the shell.
- The <newline> characters instructs the shell that the command line is ready to be processed.

```
/home/cis90/simben $ sort -r names > dogsinorder
```

*The user types in a command line followed by the **enter** key (on Macs the **return** key)*

Activity

1) Prompt
2) Parse
3) Search
4) Execute
5) Nap
6) Repeat

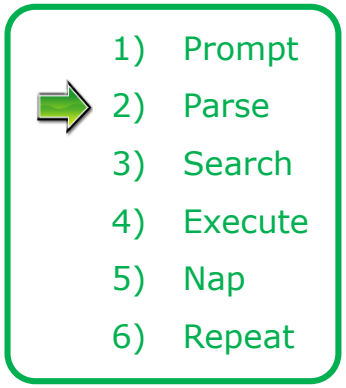
The newline character is an invisible metacharacter that triggers the shell to process the command.

1. Put five characters in a file named *five*: **echo 12345 > five**
2. Show the file contents**: cat five**
3. Show the size of your *five* file: **ls -l five**
4. Do a hex dump of your *five* file: **xxd five** and examine the output,
   hex 31 = ASCII character "1",
   hex 32 = ASCII character "2" and so forth.
   The 6th byte in the file is the newline character.

Put the size of your five file and the hex value of the newline character in the chat window.

*Optional: Use **man ascii** to check your answer.*

Shell parses what the user entered



The shell **parses** the command line entered by the user:

- The command line is carefully scanned to identify the command, options, arguments and any redirection information.
- Variables and filename expansion characters (wildcards) get processed.
- Any redirection is setup for stdin, stdout and stderr.

```
/home/cis90/simben $ sort -r names > dogsinorder
```

Parsing results: `sort` `-r` `names` `> dogsinorder`

The command is: **sort**
There is one option: **-r**
There is one argument: **names**
Redirection is: redirect **stdout** to a file named **dogsinorder**

Shell search for the command on the user's path

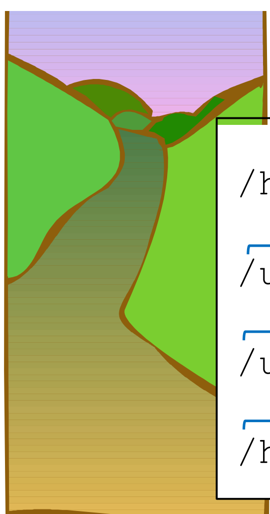
1) Prompt
2) Parse
3) Search
4) Execute
5) Nap
6) Repeat

The shell now **searches** for the command on the path:

- The path, which is an ordered list of directories, is defined by the contents of the PATH variable. Use **echo $PATH** to view.
- The shell will search in order each directory on the path to locate the command.
- If a command, such as xxxx, is not found, the shell will print:

  -bash: xxxx: command not found

- FYI, you can search for commands on the path too, like the shell does, by using the **type** command.



```
/home/cis90/simben/Poems/Yeats $ echo $PATH
        1            2          2
/usr/local/bin:/usr/bin:/usr/local/sbin:
      4              5
/usr/sbin:/home/cis90/simben/../bin:
            6            7
/home/cis90/simben/bin:.
```

The shell searches each directory on the path looking for the sort command. It finds it in the /usr/bin directory.

/usr/bin is the second directory on a CIS 90 student's path.

Activity

1) Prompt
2) Parse
3) Search
4) Execute
5) Nap
6) Repeat

Prove to yourself that the shell will find the **sort** command in the second directory on the path, /usr/*bin*.

1. Use **echo $PATH** to view your path.

2. Starting with the first directory on the path look for the sort command:
   - Use **ls -li /usr/local/bin | grep sort**
   - Use **ls -li /usr/bin | grep sort**

Write the inode number of the sort program file in the chat window.

Example

argument

option

redirection

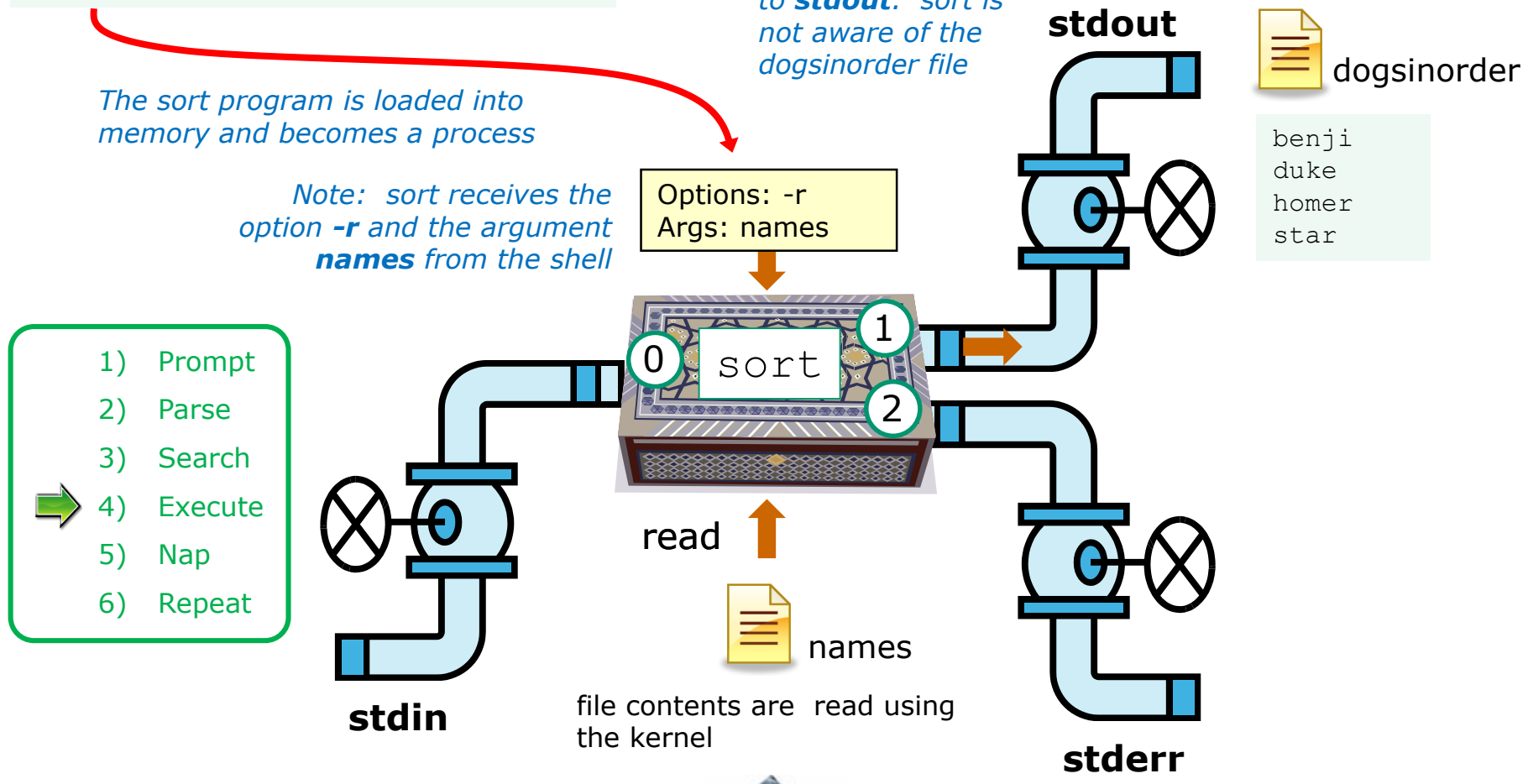
```
$ sort -r names > dogsinorder
```

*The shell connects **stdout** to the dogsinorder file*

*sort sends it's output to **stdout**. sort is not aware of the dogsinorder file*

The sort program is loaded into memory and becomes a process

*Note: sort receives the option **-r** and the argument **names** from the shell*



1) Prompt
2) Parse
3) Search
4) Execute
5) Nap
6) Repeat

Activity

1) Prompt
2) Parse
3) Search
4) Execute
5) Nap
6) Repeat

```
$ sort -r names > dogsinorder
```

What two text strings parsed by the shell were handed off to the sort command to process?

Put your answer in the chat window

Example

1) Prompt
2) Parse
3) Search
4) Execute
5) Nap
6) Repeat

While the sort process executes, the shell sleeps

Example

1) Prompt
2) Parse
3) Search
4) Execute
5) Nap
6) Repeat

When the sort process finishes the shell wakes up and starts all over again to process the next command from the user!

Subtle Differences

What is the difference between:

head -n4 letter

and

head -n4 < letter

```
/home/cis90/simben $ head -n4 letter
Hello Mother!  Hello Father!

Here I am at Camp Granada.  Things are very entertaining,
and they say we'll have some fun when it stops raining.
```

```
/home/cis90/simben $  head -n4 < letter
Hello Mother!  Hello Father!

Here I am at Camp Granada.  Things are very entertaining,
and they say we'll have some fun when it stops raining.
```

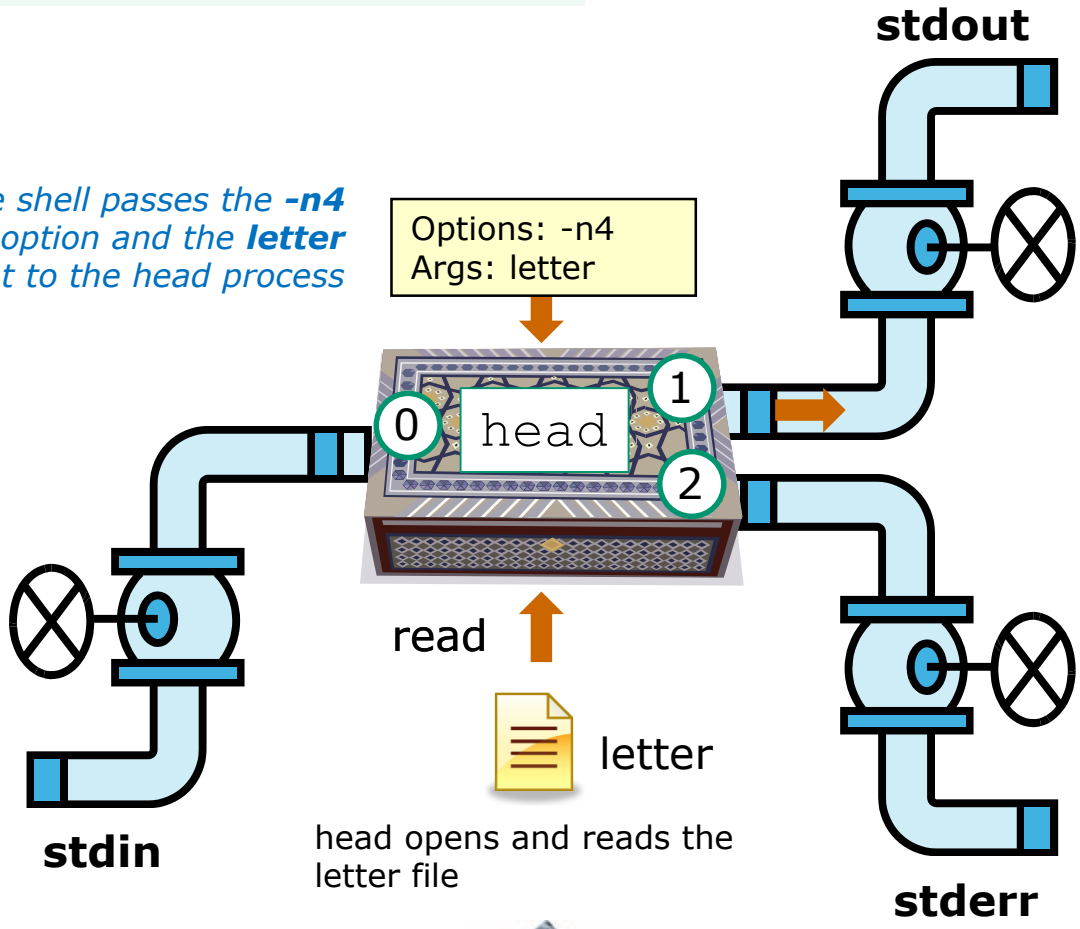
head -n4 letter

option *argument*

```
$ head -n4 letter
```

```
Hello Mother!  Hello Father!

Here I am at Camp Granada.  Things are very entertaining,
and they say we'll have some fun when it stops raining.
```

*The shell passes the **-n4** option and the **letter** argument to the head process*

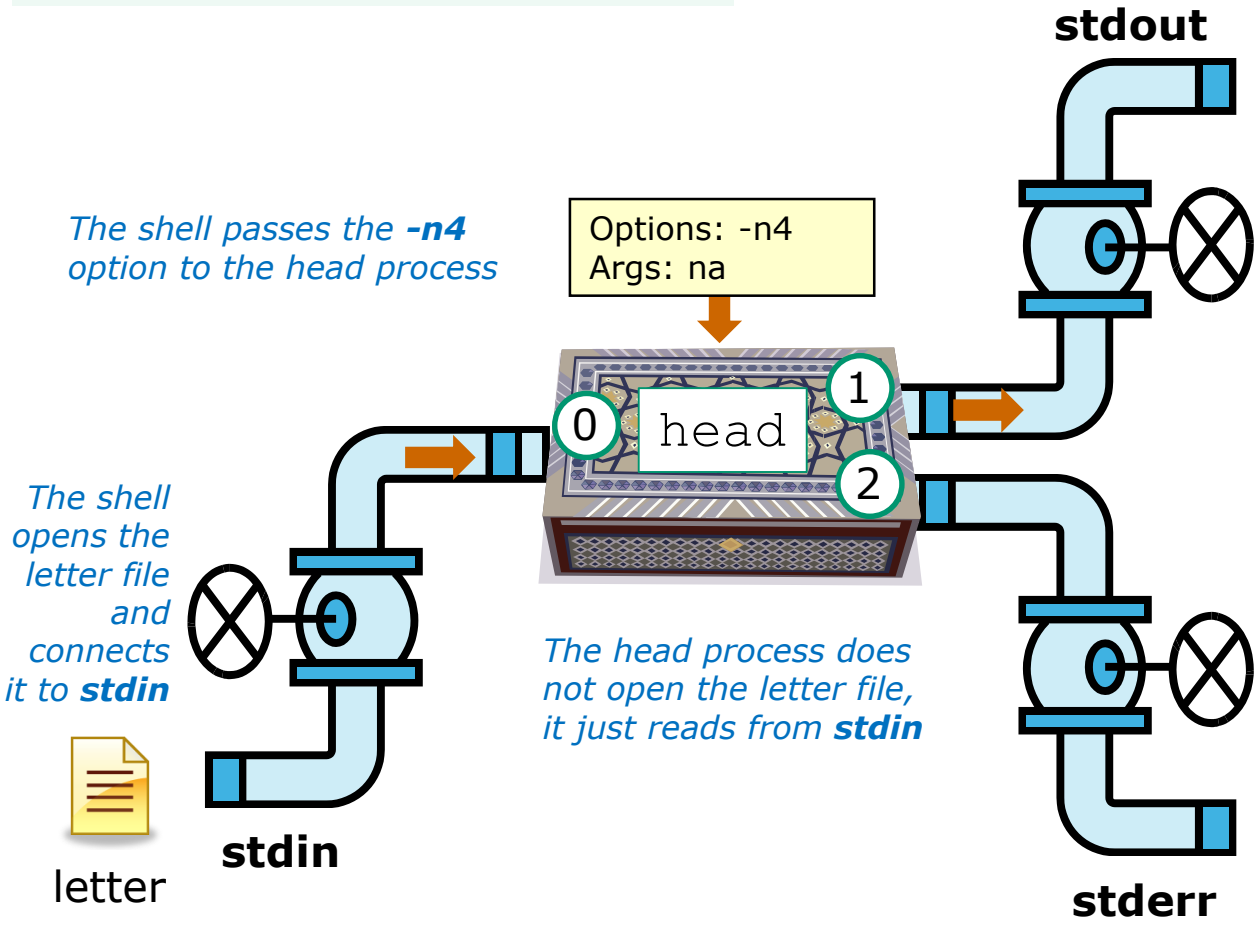


head opens and reads the letter file

head -n4 < letter

option *redirection*

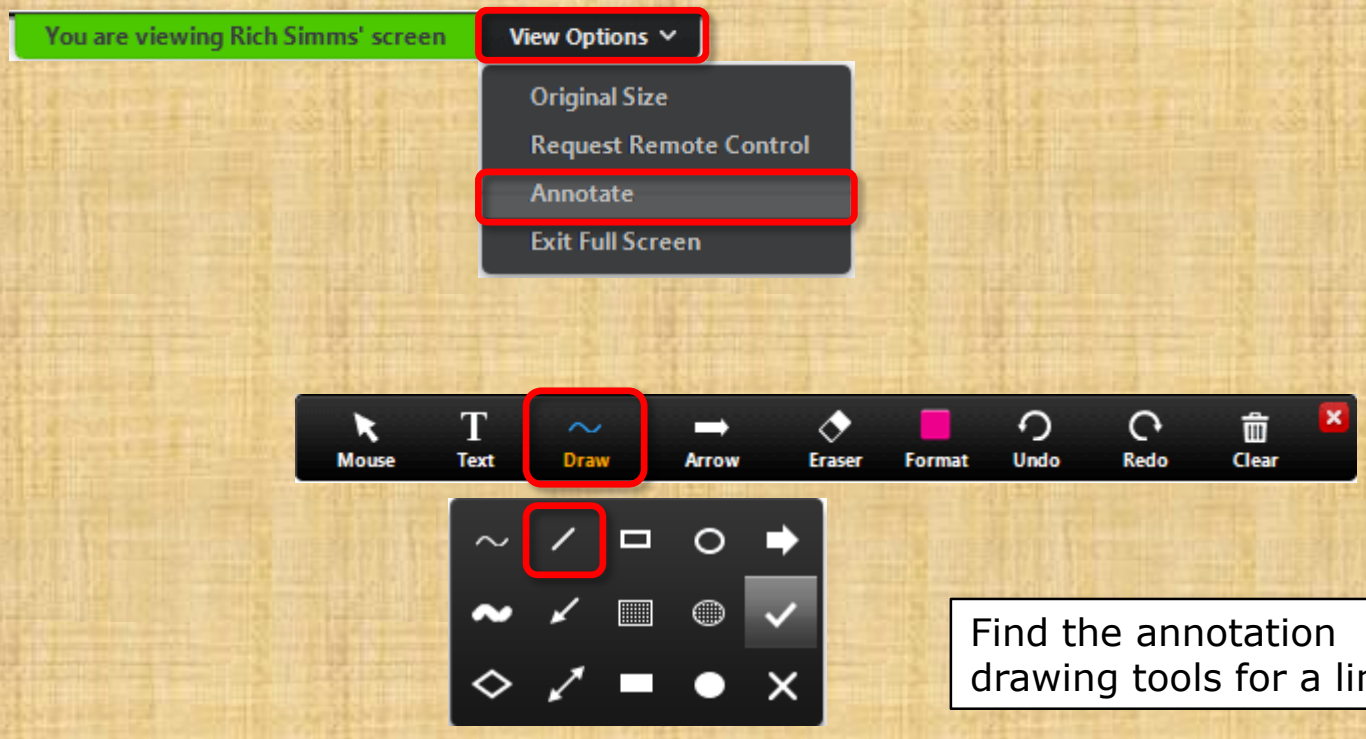
```
$ head -n4 < letter
```



Errors

Instructor: Using ConferZoom annotations

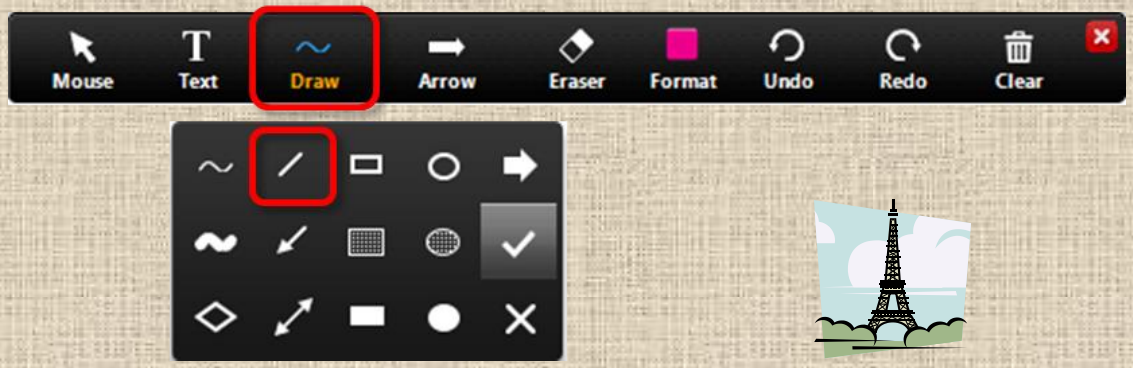
ConferZoom Annotations



Find the annotation drawing tools for a line.

View Options > Annotate Draw > "/"

CCC ConferZoom Whiteboard Activity



Select the straight line drawing tool and connect the like images

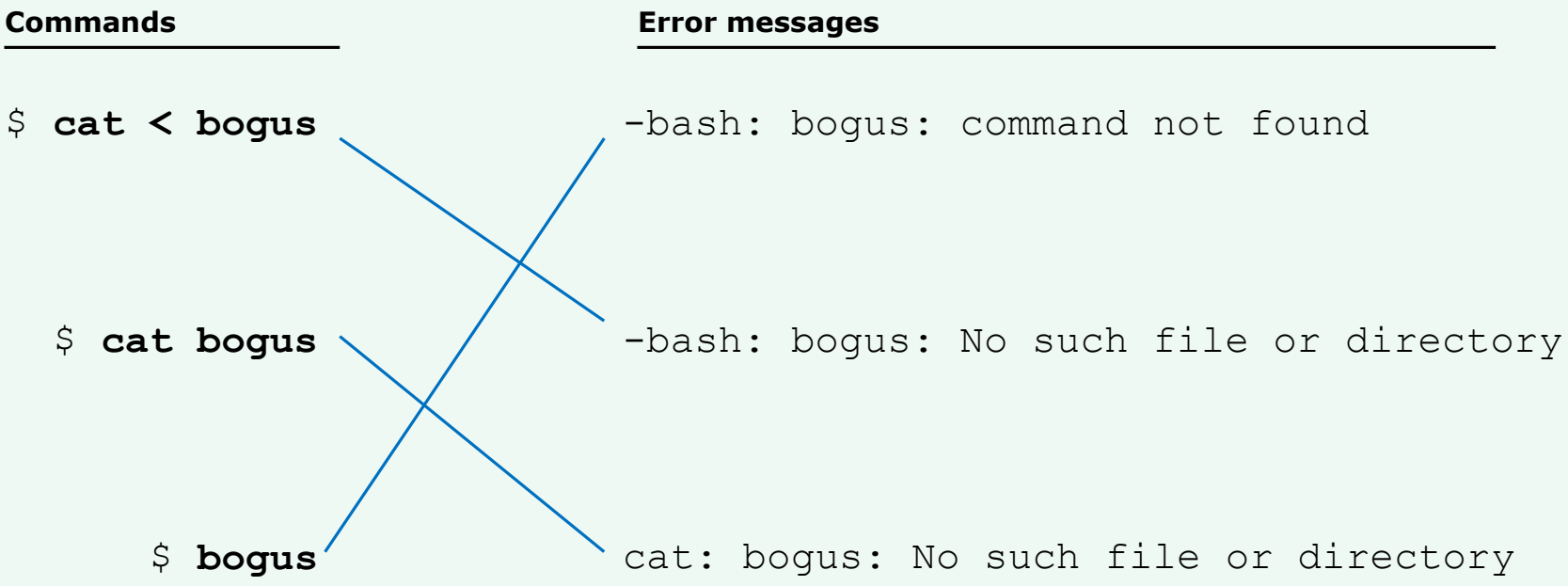
CCC Confer Whiteboard Activity

Connect with a straight line the command with the error message

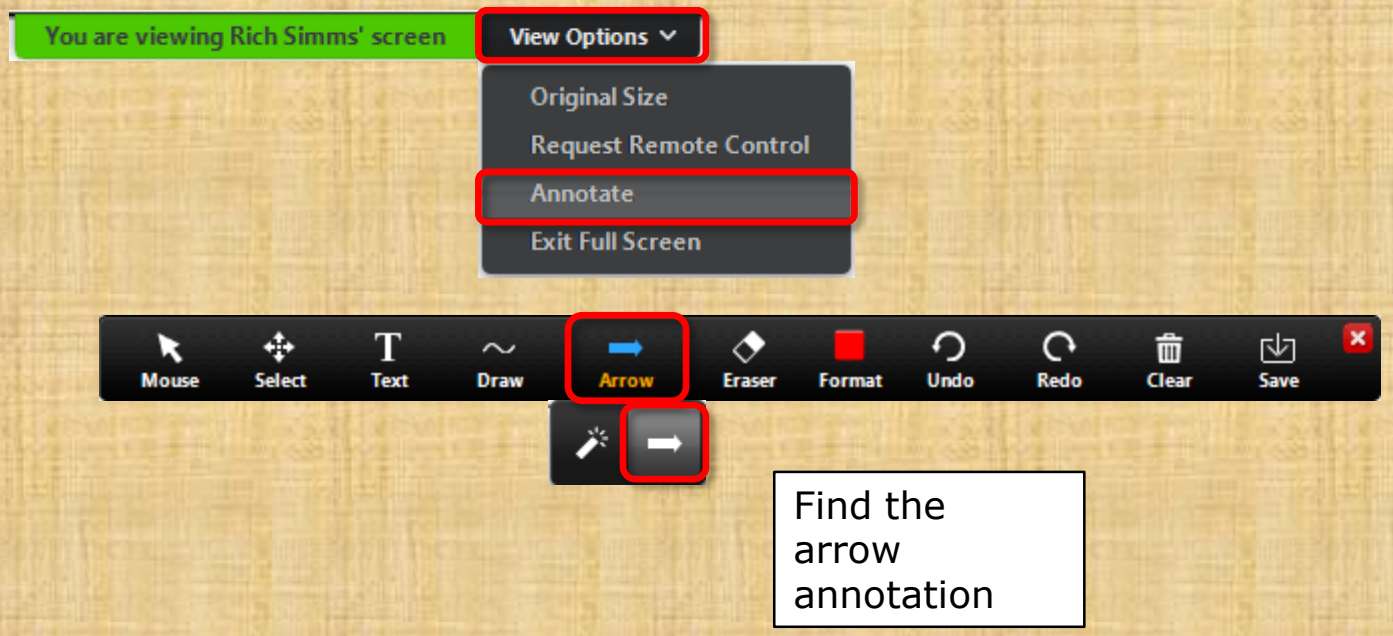
Commands	Error messages
`$ cat < bogus`	`-bash: bogus: command not found`
`$ cat bogus`	`-bash: bogus: No such file or directory`
`$ bogus`	`cat: bogus: No such file or directory`

CCC Confer Whiteboard Activity

Connect with a straight line the command with the error message



ConferZoom Annotations



View Options > Annotate Draw > Spotlight or Arrow > Arrow

CCC Confer Whiteboard Activity

Point (don't click) at the number called out by the instructor with your arrow annotation.

1

2

3

4

CCC ConferZoom Whiteboard Activity

Given: There is no file named *bogus*

```
[rsimms@oslab ~]$ cat bogus
cat: bogus: No such file or directory
```

Point to the shell step where the error message was generated

Shell Steps

1) Prompt

2) Parse

3) Search

4) Execute

5) Nap

6) Repeat

CCC ConferZoom Whiteboard Activity

Given: There is no file named *bogus*

```
[rsimms@oslab ~]$ bogus
-bash: bogus: command not found
```

Point to the shell step where the error message was generated

Shell Steps

1) Prompt

2) Parse

3) Search

4) Execute

5) Nap

6) Repeat

CCC ConferZoom Whiteboard Activity

Given: There is no file named *bogus*

```
[rsimms@oslab ~]$ cat < bogus
-bash: bogus: No such file or directory
```

Point to the shell step where the error message was generated

Shell Steps

1) Prompt

2) Parse

3) Search

4) Execute

5) Nap

6) Repeat

CCC ConferZoom Whiteboard Activity

Given: There is no file named *bogus*

```
[rsimms@oslab ~]$ bogus1 < bogus2
-bash: bogus2: No such file or directory
```

Point to the shell step where the error message was generated

Shell Steps

1) Prompt

2) Parse

3) Search

4) Execute

5) Nap

6) Repeat

CCC ConferZoom Whiteboard Activity

Given: There is no file named *bogus*

```
[rsimms@oslab ~]$ cat bogus
cat: bogus: No such file or directory
```
1) Execute

```
[rsimms@oslab ~]$ bogus
-bash: bogus: command not found
```
3) Search

```
[rsimms@oslab ~]$ cat < bogus
-bash: bogus: No such file or directory
```
2) Parse

```
[rsimms@oslab ~]$ bogus1 < bogus2
-bash: bogus2: No such file or directory
```
2) Parse

2>&1

FYI

It's descriptor clobbering time!

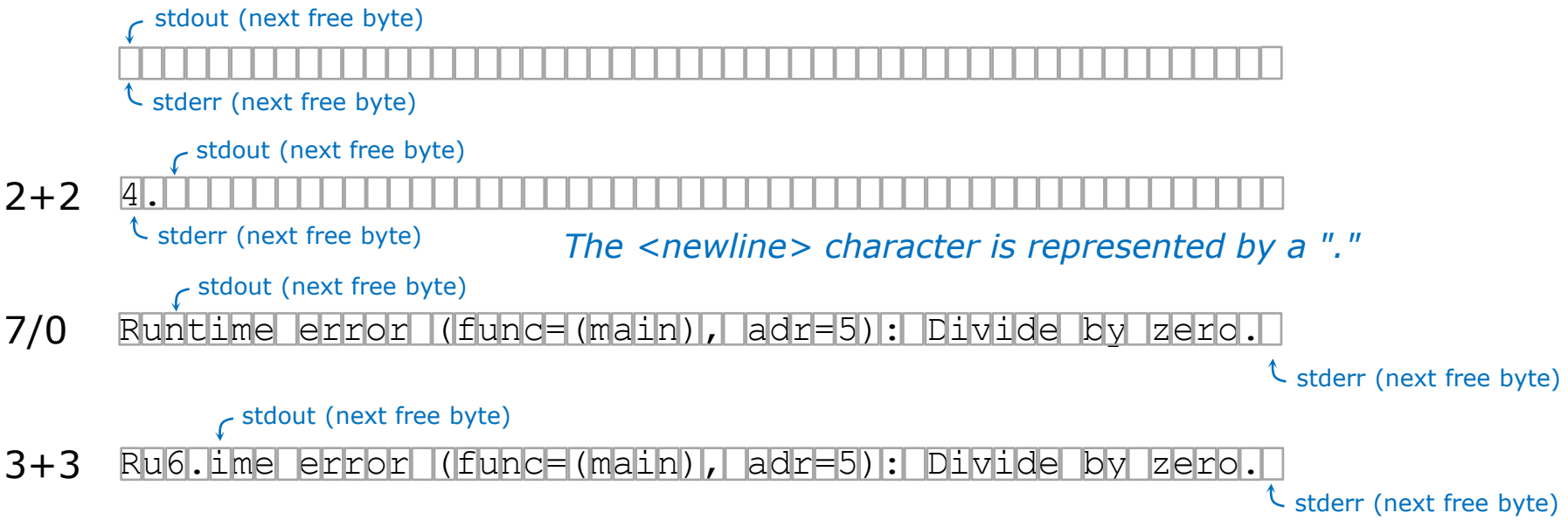
```
/home/cis90/simben $ bc > calculations 2> calculations
2+2
7/0
3+3
quit

/home/cis90/simben $ cat calculations
Ru6
ime error (func=(main), adr=5): Divide by zero
```

*Oops! Its not a good idea to redirect **stdout** and **sderr** to the same file because they clobber each other!*

FYI only

It's descriptor clobbering time!

/home/cis90/simben $ **bc > calculations 2> calculations**



/home/cis90/simben $ **cat calculations**
Ru6
ime error (func=(main), adr=5): Divide by zero

Each file descriptor keeps its own separate index into the calculations file for where to write the next line.

It's descriptor collaboration time!

stdout is redirected to calculations and stderr is redirected to the same file attached to stdout

```
/home/cis90/simben $ bc > calculations 2>&1
2+2
7/0
3+3
quit

/home/cis90/simben $ cat calculations
4
Runtime error (func=(main), adr=5): Divide by zero
6
```

*This is the correct way to redirect **stdout** and **sderr** to the same file*

More on I/O

(input/output)

programming examples

C Program I/O example
Execute the program

Running the simple program.

```
What is your name stranger? Benji
Well I'm very pleased to meet you, Benji

/home/cis90/simben $ simple > greet
What is your name stranger? Homer
/home/cis90/simben $ cat greet
Well I'm very pleased to meet you, Homer

/home/cis90/simben $ echo Duke | simple > greet
What is your name stranger? /home/cis90/simben $ cat greet
Well I'm very pleased to meet you, Duke
```

C Program I/O example
View the program

```
/home/cis90/simben/bin $ cat simple.c
char question[] = "What is your name stranger? ";
char greeting[] = "Well I'm very pleased to meet you, ";
char buffer[80];
main()
{
        int len;

        write(2, question, sizeof(question));
        len = read(0, buffer, 80);
        write(1, greeting, sizeof(greeting));
        write(1, buffer, len);
}
```

What do you think the reads and writes to 0, 1 and 2 mean in the code above?

FYI only

C Program I/O example
View the program

```
/home/cis90/simben/bin $ cat simple.c
char question[] = "What is your name stranger? ";
char greeting[] = "Well I'm very pleased to meet you, ";
char buffer[80];
main()
{
        int len;

        write(2, question, sizeof(question));
        len = read(0, buffer, 80);
        write(1, greeting, sizeof(greeting));
        write(1, buffer, len);
}
```

*Write question to **stderr***

*Read name from **stdin***

*Write greeting to **stdout***

*Write name to **stdout***

This simple program asks for a name, then responds with a greeting using the name

C Program I/O example
Compile the program

The make command is used to compile a C source text file into a binary executable

```
/home/cis90/simben/bin $ make simple
cc      simple.c    -o simple
```

Unlike a bash script, the C program source code must be compiled into a binary executable before it can be run

C Program I/O example
Execute the program

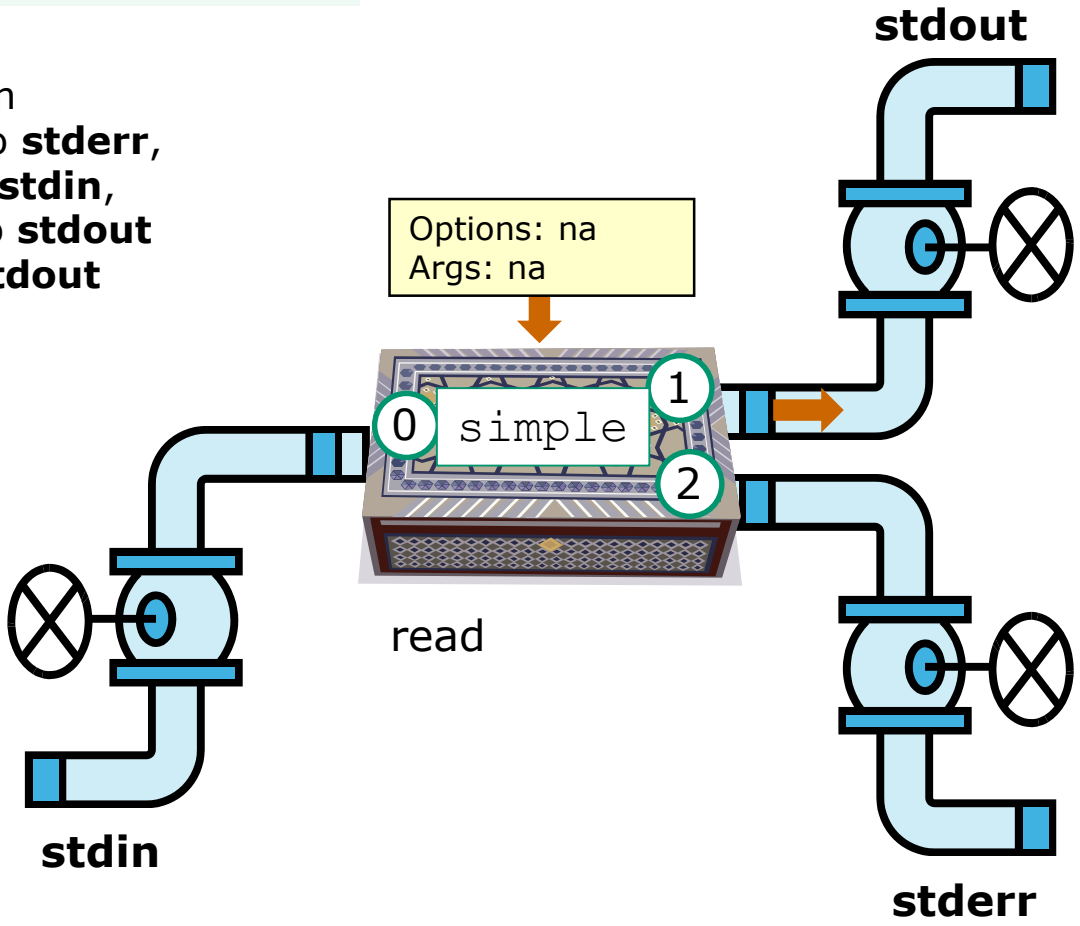
```
/home/cis90/simben/bin $ simple
What is your name stranger? Rich
Well I'm very pleased to meet you, Rich
```

Running the simple program.

C Program I/O example

```
$ simple
```

The **simple** program
1. writes question to **stderr**,
2. reads input from **stdin**,
3. writes greeting to **stdout**
4. writes name to **stdout**



3

```
Well I'm very
pleased to meet
you, Rich
```

4

2

```
Rich
```

1

```
What is your name
stranger?
```

C Program I/O example

In this example, output has been redirected to a file named myfile.

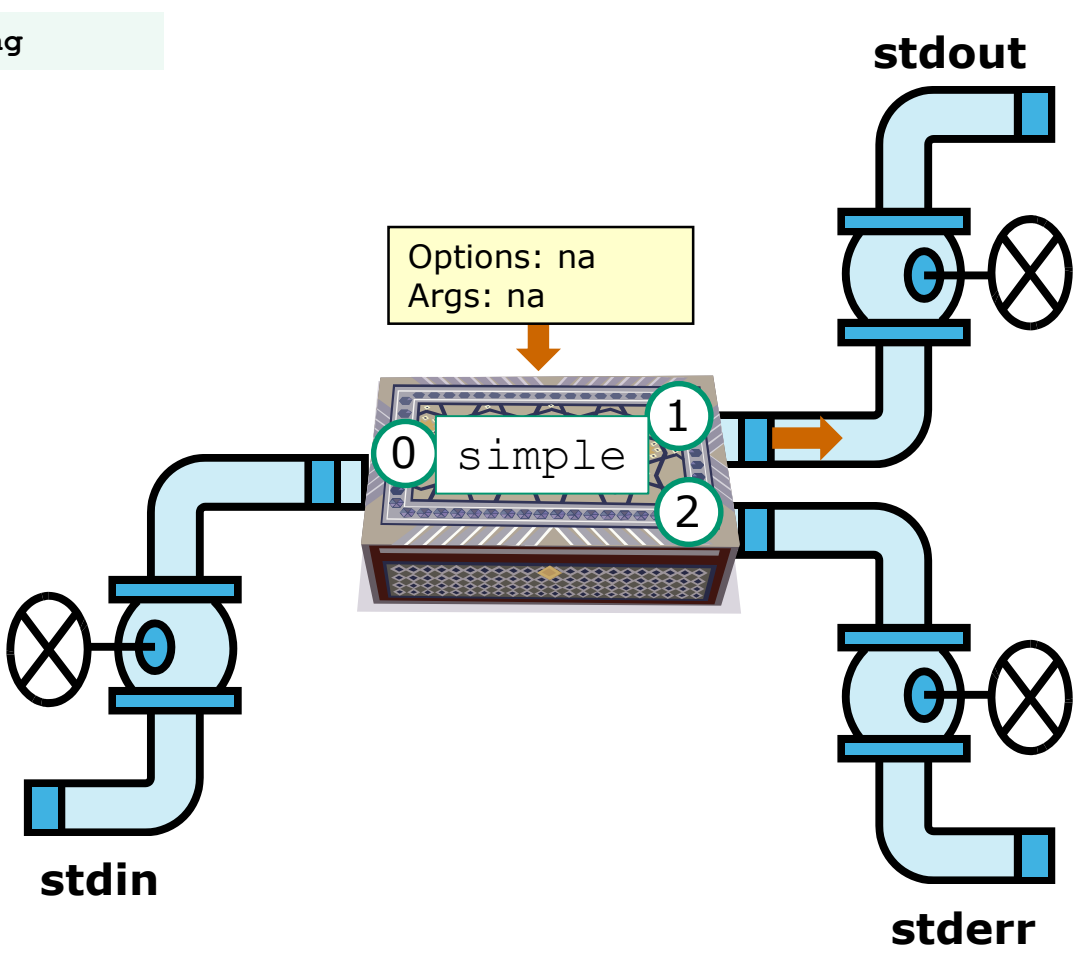
```
/home/cis90/simben/bin $ simple > myfile
What is your name stranger? Rich

/home/cis90/simben/bin $ cat myfile
Well I'm very pleased to meet you, Rich
```

*The simple program has no special knowledge (coding instructions) for a file named myfile. It just writes to **stdout** and that output will go to wherever **stdout** had been directed.*

C Program I/O example

redirection

```
$ simple > greeting
```



Activity

1. Change to your bin directory
   cd ~/bin

2. Copy the C source code from the depot directory
   cp ~/../depot/simple.c .

3. Look at your program
   cat simple.c

4. Compile the program
   make simple

5. Run the program
   simple

FYI only

C++ Program I/O example
View the program

```
/home/cis90/simben/bin $ cat simpleplus.cpp
#include <iostream>
using namespace std;

int main() {
    string question = "What is your name stranger? ";
    cerr << question;

    string buffer;
    cin >> buffer;

    string greeting = "Well I'm very pleased to meet you, ";
    cout << greeting << buffer << endl;

    return 0;
}
```

*Write question to **stderr***

*Read name from **stdin***

*Write greeting and name to **stdout***

This program is available in the depot directory

C++ Program I/O example
Compile the program

The make command is used to compile a C++ source text file into a binary executable

```
/home/cis90/simben/bin $ make simpleplus
g++      simpleplus.cpp    -o simpleplus
```

Unlike a bash script, the C++ program source code must be compiled into a binary executable before it can be run

C++ Program I/O example
Execute the program

```
/home/cis90/simben/bin $ simpleplus
What is your name stranger? Rich
Well I'm very pleased to meet you, Rich
```

Running the simpleplus program

Activity

1. Change to your bin directory
   cd ~/bin

2. Copy the C++ source code from the depot directory
   cp ~/../depot/simpleplus.cpp .

3. Look at your program
   cat simpleplus.cpp

4. Compile the program
   make simpleplus

5. Run the program
   simpleplus

Python Script I/O example
View the program

```
/home/cis90/simben $ cat simple.py
import sys
sys.stderr.write("What is your name stranger? ") Output question to stderr
name = sys.stdin.readline() Input name from stdin
sys.stdout.write("Well I'm very pleased to meet you, " + name)
```

*Output greeting and name to **stdout***

This program is available in the depot directory

Python Script I/O example
View the program

```
/home/cis90/simben $ python simple.py
What is your name stranger? Rich
Well I'm very pleased to meet you, Rich
/home/cis90/simben $
```

Running the python simple.py script

Activity

1. Change to your bin directory
   cd ~/bin

2. Copy the python script from the depot directory
   cp ~/../depot/simple.py .

3. Look at your program
   cat simple.py

4. Run the script
   python simple.py

umask

(review)

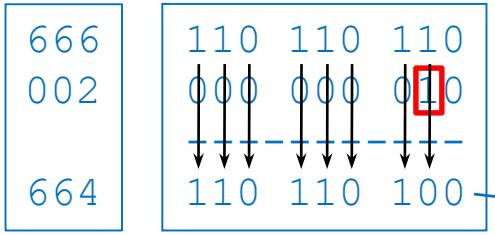
Review - applying umask bits

Example umask setting

```
/home/cis90/simben/lesson9 $ umask
0002
```

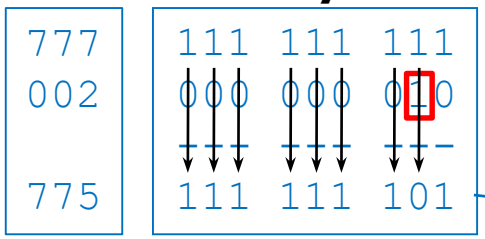
this mask indicates which permissions should NOT be set on new files or directories

New file - start with 666 and apply mask




```
/home/cis90/simben/lesson9 $ touch newfile
/home/cis90/simben/lesson9 $ ls -l newfile
-rw-rw-r-- 1 simben cis90 0 Oct 27 07:22 newfile
```

New directory - start with 777 and apply mask



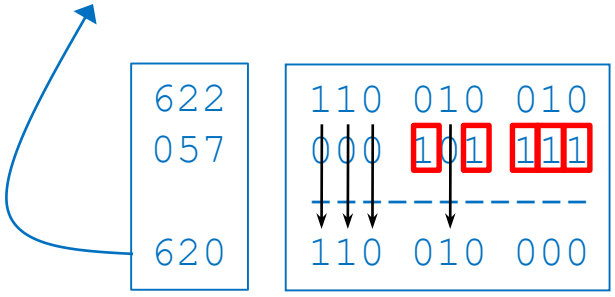

```
/home/cis90/simben/lesson9 $ mkdir newdir
/home/cis90/simben/lesson9 $ ls -ld newdir
drwxrwxr-x 2 simben cis90 4096 Oct 27 07:23 newdir
```

Any umask bits set to 1 removes the corresponding permission bit for future new files and directories

Review - Copying files

```
/home/cis90/simben $ umask 057
/home/cis90/simben $ umask
0057

/home/cis90/simben $ chmod 622 myfile
/home/cis90/simben $ cp myfile myfile.bak
/home/cis90/simben $ ls -l myfile*
-rw--w--w-. 1 simben90 cis90 0 Mar 24 17:50 myfile
-rw--w----. 1 simben90 cis90 0 Mar 24 17:51 myfile.bak
```

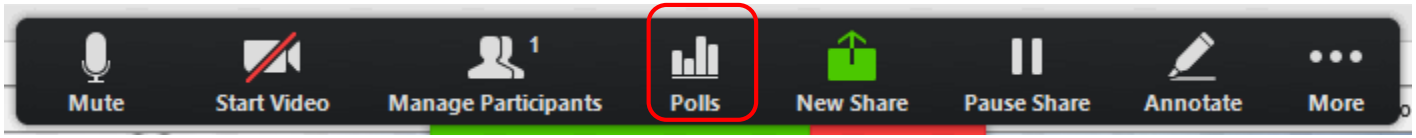
Example umask setting



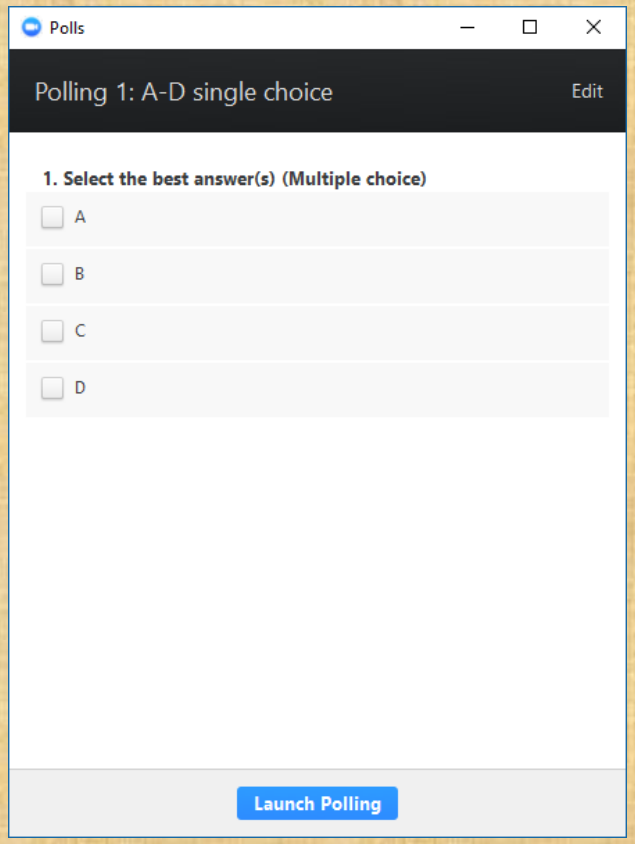
Copied file - start with original file's permissions and apply the mask

*Remember, for new files resulting from copying, instead of using the **default permissions** (666 for file and 777 for directory), use the **original file permissions** as the starting point for the mask to be applied to.*

Rich's CCC ConferZoom poll setup



Activity



Which pizza is the best?

A. Round Table
B. Pizza My Heart
C. Tony & Alba's
D. Upper Crust

Respond to the poll above

Activity

I want to change the permissions on an existing file

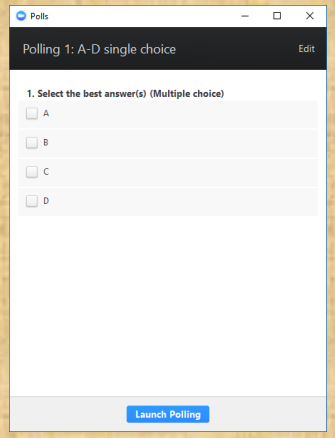
Which command does this?

A) stat

B) ls -l

C) chmod

D) umask



Respond to the poll above

Activity

I want to restrict specific permissions on files that have not been created yet

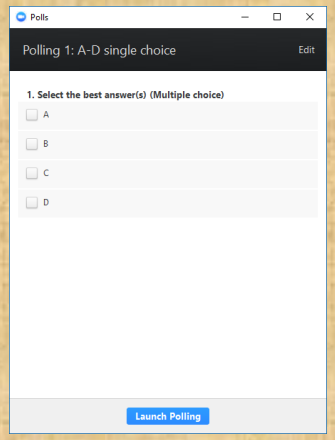
Which command does this?

A) stat

B) ls -l

C) chmod

D) umask



Respond to the poll above

Activity

I want to show the owner of a file and its permissions in mnemonic format e.g. rwxr-xr-x

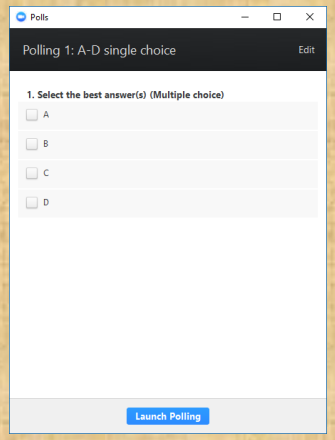
Which command does this?

A) stat

B) ls -l

C) chmod

D) umask



Respond to the poll above

Activity

I want to show the permissions on a file in numeric format e.g. 750

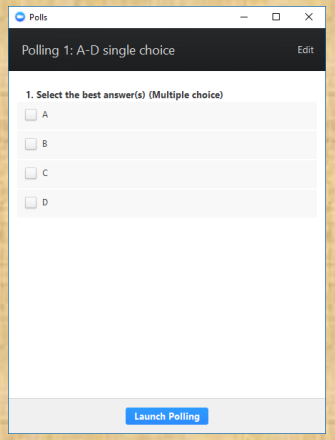
Which command does this?

A) stat

B) ls -l

C) chmod

D) umask



Respond to the poll above

More Pipeline Practice

Pipelines

Task
Record the last times Benji Simms (in CIS 90) logged in on a Monday to a file named *mylog* AND count them.

grep Benji /etc/passwd
simben76:x:1501:1076:Benji Simms:/home/cis76/simben:/bin/bash
simben90:x:1201:1090:Benji Simms:/home/cis90/simben:/bin/bash

last
last | grep simben90
last | grep simben90| grep "Mon"
last | grep simben90| grep "Mon" | tee mylog
cat mylog
last | grep simben90| grep "Mon" | tee mylog | wc -l
cat mylog

Class Exercise
Pipeline Tasks

Task
Count the last times Rich Simms was logged in on a Tuesday and record them in a file named mylog

grep "?????" /etc/passwd

last | grep ??????
last | grep ?????? | grep "Tue"
last | grep ?????? | grep "Tue" | ??? mylog
cat mylog

last | grep ?????? | grep "Tue" | ??? mylog | wc -?
cat mylog

Put your answer in the chat window.

last | grep rsimms | grep "Tue" | tee mylog | wc -l

Pipelines

Task
Print your last name as shown in /etc/passwd:

cat /etc/passwd
cat /etc/passwd | grep $LOGNAME
cat /etc/passwd | grep $LOGNAME | cut -f 5 -d ":"
cat /etc/passwd | grep $LOGNAME | cut -f 5 -d ":" | cut -f2 -d" "

Class Exercise
Pipeline Tasks

Task
What is the first name of the user milhom90?

cat /etc/passwd
cat /etc/passwd | grep ????????
cat /etc/passwd | grep ???????? | cut -f 5 -d ":"
cat /etc/passwd | grep ???????? | cut -f? -d" "

Put your answer in the chat window.

Pipelines

Task
Print a sorted list of the first names for all CIS 76 students

```
cat /etc/passwd
cat /etc/passwd | grep cis76
cat /etc/passwd | grep cis76 | cut -f 5 -d ":"
cat /etc/passwd | grep cis76 | cut -f 5 -d ":" | cut -f1 -d" "
cat /etc/passwd | grep cis76 | cut -f 5 -d ":" | cut -f1 -d" " | sort
```

Class Exercise
Pipeline Tasks

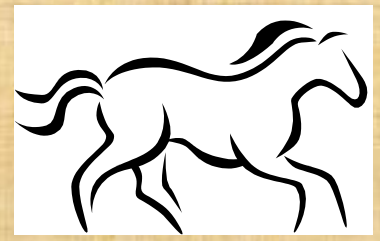
Task
Print a sorted list of the first names for CIS 90 students

cat /etc/??????
cat /etc/?????? | grep cis??
cat /etc/?????? | grep ????? | cut -f ? -d "?"
cat /etc/?????? | grep ????? | cut -f ? -d "?" | cut -f? -d"?" | ????

Put your list in the chat window.

More Review for Test 2

Review Activity



1) There is a file that contains all the user accounts on a system. The name of this file is *passwd* and it is found in the */etc* directory. Cat this file and look at it. What is the ABSOLUTE pathname of this file?

Put your answer in the chat window

Review Activity

2) In what directory does the **lspci** command reside?

Put your answer in the chat window

Review Activity

3) From your Neruda directory, what is the relative pathname to the directory containing the **lspci** command?

Put your answer in the chat window

```
/home/cis90/simben $ tree ~milhom90/trouble/
/home/cis90/milhom/trouble/
├── food
│   ├── pies
│   │   ├── apple
│   │   ├── blueberry
│   │   ├── cherry
│   │   └── mud
│   └── sandwiches
│       ├── grilled_cheese
│       ├── gyro
│       └── polish-dog
├── keep
│   └── going
│       └── down
│           └── till
│               └── the
│                   └── bottom
│                       └── names
└── rocks
    ├── igneous
    │   ├── basalt
    │   ├── diorite
    │   ├── granite
    │   └── pumice
    └── sedimentary
        ├── limestone
        ├── sandstone
        └── shale

12 directories, 15 files
/home/cis90/simben $
```

Tree diagram of Homer's trouble directory

```
tree ~milhom90/trouble/
```

To make your own trouble directory use:
cd
tar xvzf ~/../depot/trouble.tgz

Permissions on all files in Homer's trouble directory

```
/home/cis90/simben $ find ~milhom90/trouble/ -exec ls -ld {} \;
drwxr-xr-x. 5 milhom90 cis90 43 Oct 23 13:54 /home/cis90/milhom/trouble/
drwxr-xr-x. 4 milhom90 cis90 36 Oct 23 13:53 /home/cis90/milhom/trouble/food
drwxr-xr-x. 2 milhom90 cis90 61 Oct 23 13:37 /home/cis90/milhom/trouble/food/pies
-rw-r--r--. 1 milhom90 cis90 17 Oct 23 13:38 /home/cis90/milhom/trouble/food/pies/apple
-rw-r--r--. 1 milhom90 cis90 18 Oct 23 13:38 /home/cis90/milhom/trouble/food/pies/cherry
-rw-r--r--. 1 milhom90 cis90 21 Oct 23 13:38 /home/cis90/milhom/trouble/food/pies/blueberry
-rw-r--r--. 1 milhom90 cis90 15 Oct 23 13:38 /home/cis90/milhom/trouble/food/pies/mud
drw-r-xr-x. 2 milhom90 cis90 58 Oct 23 13:41 /home/cis90/milhom/trouble/food/sandwiches
-rw-r--r--. 1 milhom90 cis90 22 Oct 23 13:41 /home/cis90/milhom/trouble/food/sandwiches/grilled_cheese
-rw-r--r--. 1 milhom90 cis90 13 Oct 23 13:42 /home/cis90/milhom/trouble/food/sandwiches/gyro
-rw-r--r--. 1 milhom90 cis90 19 Oct 23 13:42 /home/cis90/milhom/trouble/food/sandwiches/polish-dog
drwxr-xr-x. 4 milhom90 cis90 40 Oct 23 13:43 /home/cis90/milhom/trouble/rocks
dr-xr-xr-x. 2 milhom90 cis90 53 Oct 23 13:45 /home/cis90/milhom/trouble/rocks/sedimentary
-rw-r--r--. 1 milhom90 cis90 0 Oct 23 13:45 /home/cis90/milhom/trouble/rocks/sedimentary/sandstone
-rw-r--r--. 1 milhom90 cis90 0 Oct 23 13:45 /home/cis90/milhom/trouble/rocks/sedimentary/shale
-rw-r--r--. 1 milhom90 cis90 0 Oct 23 13:45 /home/cis90/milhom/trouble/rocks/sedimentary/limestone
drwxr-xr-x. 2 milhom90 cis90 64 Oct 23 13:48 /home/cis90/milhom/trouble/rocks/igneous
-rw-r--r--. 1 milhom90 cis90 0 Oct 23 13:48 /home/cis90/milhom/trouble/rocks/igneous/granite
-rw-r--r--. 1 milhom90 cis90 0 Oct 23 13:48 /home/cis90/milhom/trouble/rocks/igneous/diorite
-rw-r--r--. 1 milhom90 cis90 0 Oct 23 13:48 /home/cis90/milhom/trouble/rocks/igneous/basalt
-rw-r--r--. 1 milhom90 cis90 0 Oct 23 13:48 /home/cis90/milhom/trouble/rocks/igneous/pumice
drwxr-xr-x. 3 milhom90 cis90 19 Oct 23 13:54 /home/cis90/milhom/trouble/keep
drwxr-xr-x. 3 milhom90 cis90 18 Oct 23 13:54 /home/cis90/milhom/trouble/keep/going
drwxr-xr-x. 3 milhom90 cis90 18 Oct 23 13:54 /home/cis90/milhom/trouble/keep/going/down
drwxr-xr-x. 3 milhom90 cis90 17 Oct 23 13:54 /home/cis90/milhom/trouble/keep/going/down/till
drwxr-xr-x. 3 milhom90 cis90 20 Oct 23 13:54 /home/cis90/milhom/trouble/keep/going/down/till/the
drwxr-xr-x. 2 milhom90 cis90 19 Oct 23 13:55 /home/cis90/milhom/trouble/keep/going/down/till/the/bottom
-rw-r--r--. 1 milhom90 cis90 27 Oct 23 13:55 /home/cis90/milhom/trouble/keep/going/down/till/the/bottom/names
/home/cis90/simben $
```

`find ~milhom90/trouble/ -exec ls -ld {} \;`

Review Activity

Homer

```
/home/cis90/milhom $ ls -l trouble/food/sandwiches/
ls: cannot access trouble/food/sandwiches/grilled_cheese: Permission denied
ls: cannot access trouble/food/sandwiches/gyro: Permission denied
ls: cannot access trouble/food/sandwiches/polish-dog: Permission denied
total 0
-????????? ? ? ? ?            ? grilled_cheese
-????????? ? ? ? ?            ? gyro
-????????? ? ? ? ?            ? polish-dog
```

Benji

```
/home/cis90/simben $ ls -l ~milhom90/trouble/food/sandwiches/
total 12
-rw-r--r--. 1 milhom90 cis90 22 Oct 23 13:41 grilled_cheese
-rw-r--r--. 1 milhom90 cis90 13 Oct 23 13:42 gyro
-rw-r--r--. 1 milhom90 cis90 19 Oct 23 13:42 polish-dog
```

4) Homer is getting a weird long listing on his *sandwiches* directory. Benji isn't. How can Homer fix this?

Put your answer in the chat window

Review Activity

Homer

```
/home/cis90/milhom/trouble/rocks/sedimentary $ touch siltstone
touch: cannot touch 'siltstone': Permission denied
/home/cis90/milhom/trouble/rocks/sedimentary $
```

5) Homer can't add any more rock files to his *sedimentary* directory? How can he fix this?

Put your answer in the chat window

Review Activity Answers

1) /etc/passwd

2) /usr/sbin

3) ../../../../../usr/bin

4) Homer does not have execute permission for user on his sandwiches directory:

```
/home/cis90/milhom $ ls -ld trouble/food/sandwiches/
drw-r-xr-x. 2 milhom90 cis90 58 Oct 23 13:41 trouble/food/sandwiches/
```

Homer can fix using: **chmod u+x trouble/food/sandwiches/**

5) Homer does not have write permission for user on his sedimentary directory:

```
/home/cis90/milhom $ ls -ld trouble/rocks/sedimentary/
dr-xr-xr-x. 2 milhom90 cis90 53 Oct 23 13:45 trouble/rocks/sedimentary/
```

Homer can fix using: **chmod u+w trouble/rocks/sedimentary/**

More on pipelines

Not all commands are filters
(filters read from stdin and write to stdout)

*The **wc** command is a filter.*

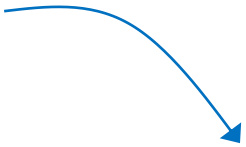
```
/home/cis90/simben $ head -n2 poems/Anon/nursery
Jack and Jill went up the hill
to fetch a pail of water.
/home/cis90/simben $ head -n2 poems/Anon/nursery | wc -l
2
/home/cis90/simben $
```

*But the **echo** command isn't (doesn't read from **stdin**)*

```
/home/cis90/simben $ head -n2 poems/Anon/nursery | echo

Oops .... this doesn't work!
/home/cis90/simben $
```

xargs command

xargs *to the rescue!*



```
/home/cis90/simben $ head -n2 poems/Anon/nursery | xargs echo
Jack and Jill went up the hill to fetch a pail of water.
```

*The **xargs** command will read **stdin** and call another command using the input as the arguments.*

Another example

Why can't Benji make a banner using the output of the date command?

```
/home/cis90/simben $ date | banner
Enter a string of up to 10 characters.
/home/cis90/simben $
```

huh? Oh, this is what banner prints when it receives no arguments on the command line

Because banner is not a filter and does not read from stdin!

Another example

```
/home/cis90/simben $ date | xargs banner
#     # ####### #     #
##   ## #     # ##    #
# # # # #     # # #   #
#  #  # #     # #  #  #
#     # #     # #   # #
#     # #     # #    ##
#     # ####### #     #

#######  #####  #######
#     # #     #    #
#     # #          #
#     # #          #
#     # #          #
#     # #     #    #
#######  #####     #

 #####   #####
#     # #     #
      #       #
 #####   #####
#       #
#       #
####### #######

   #       #             #####   #####          #######  #####
  ##      ##      ###   #     # #     #   ###   #       #     #
 # #     # #      ###         # #         ###   #       #     #
   #       #             #####  ######         ######   #####
   #       #      ###   #       #     #   ###         # #     #
   #       #      ###   #       #     #   ###   #     # #     #
 #####   #####          #######  #####          #####   #####

######  ######  #######
#     # #     #    #
#     # #     #    #
######  #     #    #
#       #     #    #
#       #     #    #
#       ######     #

 #####    ###      #     #####
#     #  #   #    ##    #     #
      # #     #  # #          #
 #####  #     #    #     #####
#       #     #    #    #
#        #   #     #    #
#######   ###    #####  #######
```

xargs *to the rescue again!*

Not all commands are filters
(filters read from stdin and write to stdout)

*The **ls** command does not read from **stdin** either*

```
/home/cis90/simben $ find poems -type d
poems
poems/Shakespeare
poems/Yeats
poems/Anon
poems/Blake


/home/cis90/simben $ find poems -type d | ls -ld
drwxr-xr-x. 18 simben90 cis90 4096 Oct 22 09:49 .
/home/cis90/simben $
```

Benji was hoping that he could get a long listing of his poems directory and all its sub-directories. Instead he gets a long listing of his home directory!

Not all commands are filters
(filters read from stdin and write to stdout)


```
/home/cis90/simben $ find poems -type d | xargs ls -ld
drwxr-xr-x. 6 simben90 cis90 4096 Oct 20 15:06 poems
drwxr-xr-x. 2 simben90 cis90 4096 Oct  5 10:26 poems/Anon
drwxr-xr-x. 2 simben90 cis90 4096 Oct 20 15:06 poems/Blake
drwxr-xr-x. 2 simben90 cis90 4096 Oct 20 15:06 poems/Shakespeare
drwxr-xr-x. 2 simben90 cis90 4096 Oct 20 15:06 poems/Yeats
/home/cis90/simben $
```

*The **ls** command is not a filter so it does not read from **stdin***

***xargs** reads the names of the files found by the **find** command and uses them as arguments on the **ls -ld** command*

Not all commands are filters
(filters read from stdin and write to stdout)

```
/home/cis90/simben $ find poems -type d -exec ls -ld {} \;
drwxr-xr-x. 6 simben90 cis90 4096 Oct 20 15:06 poems
drwxr-xr-x. 2 simben90 cis90 4096 Oct 20 15:06 poems/Shakespeare
drwxr-xr-x. 2 simben90 cis90 4096 Oct 20 15:06 poems/Yeats
drwxr-xr-x. 2 simben90 cis90 4096 Oct  5 10:26 poems/Anon
drwxr-xr-x. 2 simben90 cis90 4096 Oct 20 15:06 poems/Blake
/home/cis90/simben $
```

*The **find** command also has a **-exec** option that will run a command on what is found. The **{}** represent the arguments which are names of files found by the **find** command.*

Things that Hide

Memorize This

How to locate files:

To locate by name ... use **find**

To locate by user, type, group, etc[1]. ... use **find**

To locate by contents[2] within specific files ... use **grep**

To locate by contents[2] in a branch of the file tree ... use recursive **grep**

1 File inode properties (e.g. shown in a long listing)
2 File data (e.g. shown when catting a file)

Finding Things

Task
Find all files in the */usr/share/doc* branch of the file tree that are named "BUGS"

```
find /usr/share/doc -name "BUGS"
```

```
/usr/share/doc/ppl-0.10.2/BUGS
/usr/share/doc/ltrace-0.5/BUGS
/usr/share/doc/perl-IO-Socket-SSL-1.31/BUGS
/usr/share/doc/glibc-2.12/BUGS
/usr/share/doc/parted-2.1/BUGS
/usr/share/doc/cvs-1.11.23/BUGS
/usr/share/doc/patchutils-0.3.1/BUGS
/usr/share/doc/procps-3.2.8/BUGS
/usr/share/doc/gettext-0.17/BUGS
/usr/share/doc/curl-7.19.7/BUGS
/usr/share/doc/sed-4.2.1/BUGS
/usr/share/doc/SDL-1.2.14/BUGS
/usr/share/doc/cairo-1.8.8/BUGS
/usr/share/doc/emacs-common-23.1/BUGS
/usr/share/doc/tcsh-6.17/BUGS
/usr/share/doc/unzip-6.0/BUGS
/usr/share/doc/vsftpd-2.2.2/BUGS
/usr/share/doc/dejavu-fonts-common-2.30/BUGS
/usr/share/doc/nano-2.0.9/BUGS
[rsimms@oslab ~]$
```

Use find to search for files by name, type, user, group, etc.

Finding Things

Task
Count all the files in the */home* branch of the file tree that are owned by rsimms. Discard any permission errors.

```
find /???? -user ?????? 2> /dev/???? | ?? -l
```

Write your answer in the chat window

/home/cis90/simben $ find /home -user rsimms 2> /dev/null | wc -l

Finding Things

Task

Find all files in the */home/cis90/bin that are regular files and belong to the staff group.*

```
find /home/cis90/bin -group staff -type f
```

```
/home/cis90/bin/enlightenment
/home/cis90/bin/allscripts
/home/cis90/bin/list
/home/cis90/bin/submit.sp15.v1
/home/cis90/bin/tinsam90/schedule.pyc
/home/cis90/bin/tinsam90/schedule.py
/home/cis90/bin/tinsam90/forums.py
/home/cis90/bin/tinsam90/tips.py
/home/cis90/bin/tinsam90/grade.py
/home/cis90/bin/submitx
/home/cis90/bin/old/submit.fa14.v5
/home/cis90/bin/old/checkgrades.py.fa14
/home/cis90/bin/old/allscripts.sp14
/home/cis90/bin/old/check10.v2
/home/cis90/bin/old/submit.fa14.v1
/home/cis90/bin/old/check10.v1
/home/cis90/bin/old/submit.fa14.v4
/home/cis90/bin/old/checkgrades.py.sp14
/home/cis90/bin/old/submit.fa14.v2
/home/cis90/bin/old/submit.fa14.v3
/home/cis90/bin/old/submit.fa14.v6
/home/cis90/simben $
```

Use find to search for files by name, type, user, group, etc.

Finding Things

Task
Count all the directories in the */home/cis90* branch of the file tree that belong to the cis90 group. Discard any permission errors.

```
???? /home/????? -type ? -group ????? ?? /dev/null | ?? -?
```

Write your answer in the chat window

find /home/cis90 -type d -group cis90 2> /dev/null | wc -l

Finding Things

Task
Find an account for simben90 in /etc/passwd

```
/home/cis90/simben $ grep simben90 /etc/passwd
simben90:x:1201:190:Benji Simms:/home/cis90/simben:/bin/bash
```

Finding Things

Task
Find your account in /etc/passwd

```
grep ??????90 /etc/passwd
```

Write your results in the chat window

grep $LOGNAME /etc/passwd

Finding Things

Task

Find all files in the */usr/src* branch of the file tree that contain "Torvalds"

```
grep -r "Torvalds" /usr/src
```

```
/usr/src/kernels/2.6.32-220.23.1.el6.i686/arch/powerpc/Makefile:# Copyright (C) 1994 by Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/arch/s390/Makefile:# Copyright (C) 1994 by Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/arch/parisc/Makefile:# Copyright (C) 1994 by Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/arch/alpha/Makefile:# Copyright (C) 1994 by Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/arch/alpha/boot/Makefile:# Copyright (C) 1994 by Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/arch/x86/include/asm/i387.h: * Copyright (C) 1994 Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/arch/x86/include/asm/irq.h: *  (C) 1992, 1993 Linus Torvalds, (C) 1997 Ingo Molnar
/usr/src/kernels/2.6.32-220.23.1.el6.i686/arch/x86/include/asm/thread_info.h: * - Incorporating suggestions made by Linus Torvalds and Dave Miller
/usr/src/kernels/2.6.32-220.23.1.el6.i686/arch/x86/include/asm/bitops.h: * Copyright 1992, Linus Torvalds.
/usr/src/kernels/2.6.32-220.23.1.el6.i686/arch/x86/include/asm/stacktrace.h: * Copyright (C) 1991, 1992  Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/arch/x86/include/asm/hw_irq.h: *  (C) 1992, 1993 Linus Torvalds, (C) 1997 Ingo Molnar
/usr/src/kernels/2.6.32-220.23.1.el6.i686/arch/x86/include/asm/delay.h: * Copyright (C) 1993 Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/arch/x86/include/asm/sync_bitops.h: * Copyright 1992, Linus Torvalds.
/usr/src/kernels/2.6.32-220.23.1.el6.i686/arch/x86/boot/Makefile:# Copyright (C) 1994 by Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/include/linux/ioport.h: * Authors:    Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/include/linux/pagemap.h: * Copyright 1995 Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/include/linux/thread_info.h: * - Incorporating suggestions made by Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/include/linux/ext2_fs.h: *  Copyright (C) 1991, 1992  Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/include/linux/nilfs2_fs.h: *  Copyright (C) 1991, 1992  Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/include/linux/dcache.h: * with heavy changes by Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/include/linux/completion.h: * (C) Copyright 2001 Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/include/linux/ext3_fs.h: *  Copyright (C) 1991, 1992  Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/include/linux/ide.h: *  Copyright (C) 1994-2002  Linus Torvalds & authors
/usr/src/kernels/2.6.32-220.23.1.el6.i686/include/linux/delay.h: * Copyright (C) 1993 Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/include/linux/ext3_fs_sb.h: *  Copyright (C) 1991, 1992  Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/include/linux/ext3_fs_i.h: *  Copyright (C) 1991, 1992  Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/include/linux/ext2_fs_sb.h: *  Copyright (C) 1991, 1992  Linus Torvalds
/usr/src/kernels/2.6.32-220.23.1.el6.i686/include/asm-generic/tlb.h:  * Based on code from mm/memory.c Copyright Linus Torvalds and others.
/usr/src/kernels/2.6.32-220.23.1.el6.i686/scripts/package/builddeb:Copyright:  1991 - 2009 Linus Torvalds and others.
/usr/src/kernels/2.6.32-220.23.1.el6.i686/scripts/get_maintainer.pl:push(@penguin_chief,"Linus Torvalds:torvalds\@linux-foundation.org");
/usr/src/kernels/2.6.32-220.23.1.el6.i686/scripts/checkstack.pl:#      Inspired by Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/arch/powerpc/Makefile:# Copyright (C) 1994 by Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/arch/s390/Makefile:# Copyright (C) 1994 by Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/arch/parisc/Makefile:# Copyright (C) 1994 by Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/arch/alpha/Makefile:# Copyright (C) 1994 by Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/arch/alpha/boot/Makefile:# Copyright (C) 1994 by Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/arch/x86/include/asm/i387.h:  * Copyright (C) 1994 Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/arch/x86/include/asm/irq.h:  *      (C) 1992, 1993 Linus Torvalds, (C) 1997 Ingo Molnar
/usr/src/kernels/2.6.32-71.el6.i686/arch/x86/include/asm/thread_info.h: * - Incorporating suggestions made by Linus Torvalds and Dave Miller
/usr/src/kernels/2.6.32-71.el6.i686/arch/x86/include/asm/bitops.h:  * Copyright 1992, Linus Torvalds.
/usr/src/kernels/2.6.32-71.el6.i686/arch/x86/include/asm/hw_irq.h:  *  (C) 1992, 1993 Linus Torvalds, (C) 1997 Ingo Molnar
/usr/src/kernels/2.6.32-71.el6.i686/arch/x86/include/asm/delay.h:  * Copyright (C) 1993 Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/arch/x86/include/asm/sync_bitops.h:  * Copyright 1992, Linus Torvalds.
/usr/src/kernels/2.6.32-71.el6.i686/arch/x86/boot/Makefile:# Copyright (C) 1994 by Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/include/linux/ioport.h:  * Authors:  Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/include/linux/pagemap.h:  * Copyright 1995 Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/include/linux/thread_info.h:  * - Incorporating suggestions made by Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/include/linux/ext2_fs.h:  *  Copyright (C) 1991, 1992  Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/include/linux/nilfs2_fs.h:  *  Copyright (C) 1991, 1992  Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/include/linux/dcache.h:  * with heavy changes by Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/include/linux/completion.h:  * (C) Copyright 2001 Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/include/linux/ext3_fs.h:  *  Copyright (C) 1991, 1992  Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/include/linux/ide.h:  *  Copyright (C) 1994-2002  Linus Torvalds & authors
/usr/src/kernels/2.6.32-71.el6.i686/include/linux/delay.h:  * Copyright (C) 1993 Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/include/linux/ext3_fs_sb.h:  *  Copyright (C) 1991, 1992  Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/include/linux/ext3_fs_i.h:  *  Copyright (C) 1991, 1992  Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/include/linux/ext2_fs_sb.h:  *  Copyright (C) 1991, 1992  Linus Torvalds
/usr/src/kernels/2.6.32-71.el6.i686/include/asm-generic/tlb.h:  * Based on code from mm/memory.c Copyright Linus Torvalds and others.
/usr/src/kernels/2.6.32-71.el6.i686/scripts/package/builddeb:Copyright:  1991 - 2009 Linus Torvalds and others.
/usr/src/kernels/2.6.32-71.el6.i686/scripts/get_maintainer.pl:push(@penguin_chief,"Linus Torvalds:torvalds\@linux-foundation.org");
/usr/src/kernels/2.6.32-71.el6.i686/scripts/checkstack.pl:#      Inspired by Linus Torvalds
[rsimms@oslab ~]$
```

*Do a recursive grep to search the **contents** of files in an entire branch of the file tree.*

Finding Things

Task
Count the number of files in the */usr/src* branch of the file tree that contain "Stallman"

```
grep -? "Stallman" /???/??? | wc -?
```

Write your answer in the chat window

grep -r "Stallman" /usr/src | wc -l

Eggs, Treats and Tricks

trick or treat

A number of *trick* and *treat* files have been distributed within your home directory and sub-directories!

1. Can you find them? There should be an obvious one in your home directory. The rest are scattered in the various subdirectories you own.

2. Make a new directory named *bag* in your home directory and see how many *trick* or *treat* files you can move into it.

For the first several trick or treat files you find, cat the file, and write in the chat window what you got.

Then write in the chat window when you get all six treats and then again when you get all 12 tricks and treats.

Instructor: sudo /home/rsimms/cis90/halloween/hidetreats

Review

Jim's Summary Pages

Jim has some really good summary information on Lessons 6-8 on his web site:

Lesson 6 - Managing Files
https://web.archive.org/web/20100708145536/http://www.cabrillo.edu/~jgriffin/CIS90/files/lecture5.html

Lesson 7 - File Permissions
https://web.archive.org/web/20100708151130/http://www.cabrillo.edu/~jgriffin/CIS90/files/lecture6.html

Lesson 8 - Input/Output Processing
https://web.archive.org/web/20100708151725/http://www.cabrillo.edu/~jgriffin/CIS90/files/lecture7.html

Make Teams

Make Teams

Everyone needs to be on ConferZoom today

Instructor:
PS1="[\u@\h \W] \t \$ "
/home/rsimms/scripts/teams (use teams alias)

Flashcard Practice

Flashcards

Flashcards
L6=20
L7=15
L8=16

Rules

- Chat window belongs to team that is up
- Team gets the point if anyone on the team writes a correct answer in the chat window in 15 seconds

Instructor timer:
i=15; while [$i -gt 0]; do clear; banner $i; let i=i-1; sleep 1; done; clear; banner done
(Use **countdown** alias)

Flash Cards

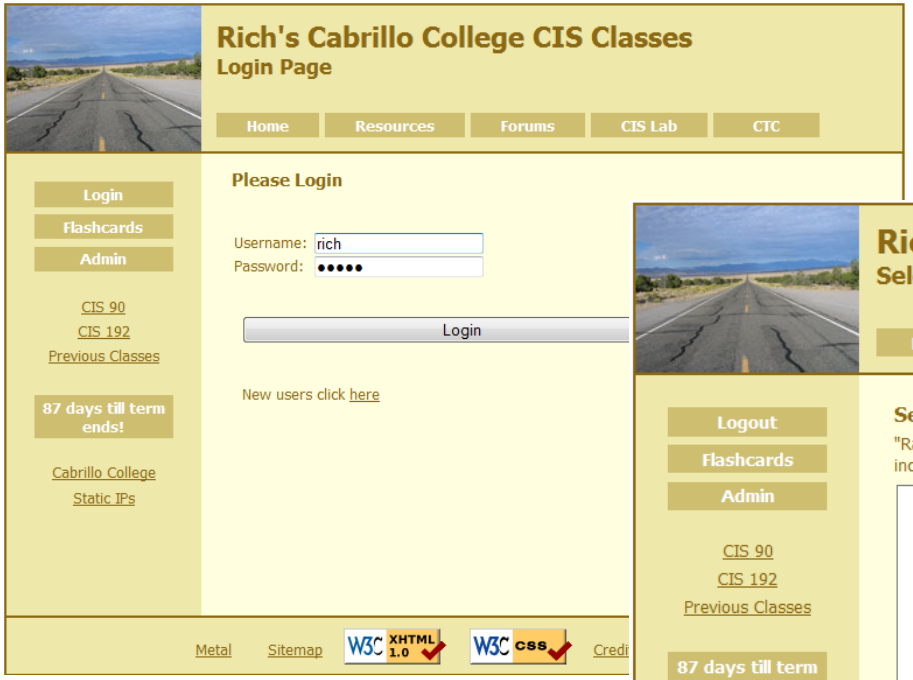
Click on Flashcards in left panel

Register if this is the first time using Flashcards

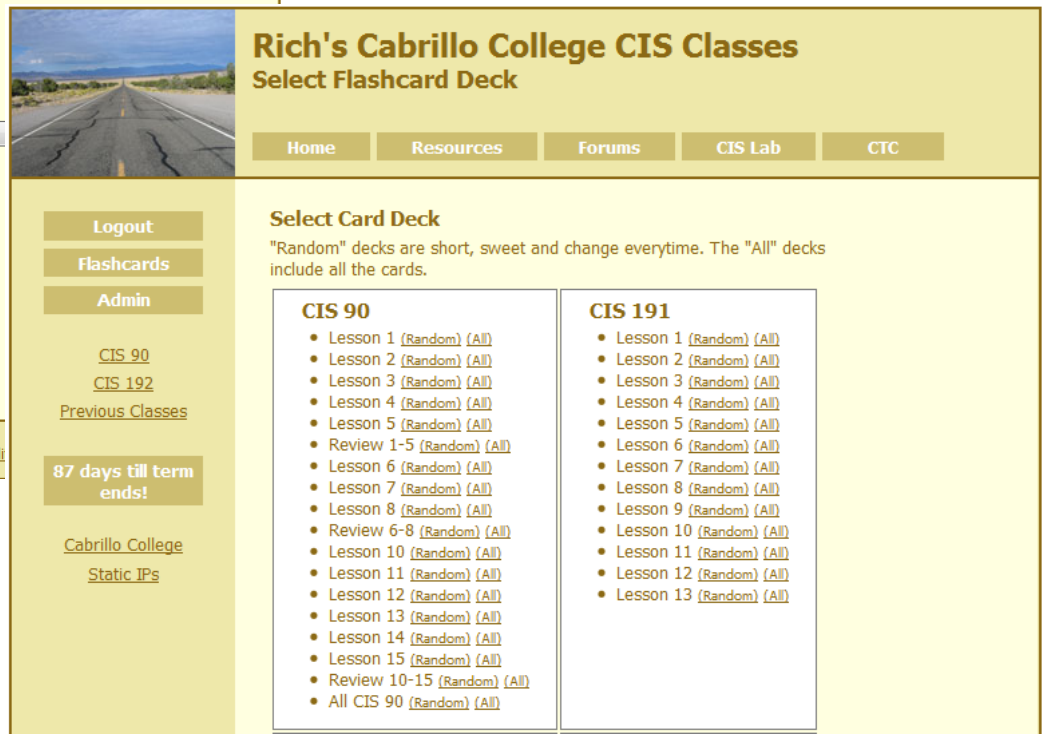
Register and choose a username and password of your choice

Logging in and using Flashcards

Login with your username and password



Select deck of cards



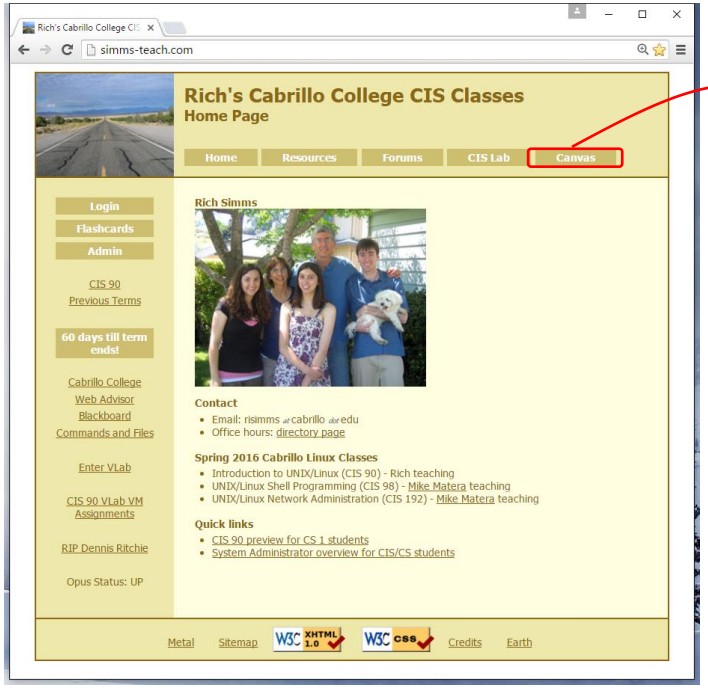
Class Exercise
Flashcards

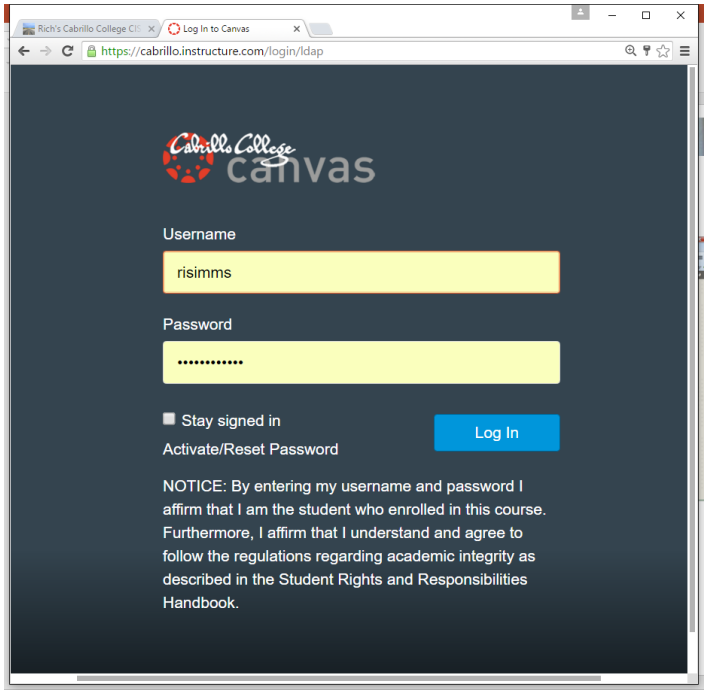
- Browse to simms-teach.com
- Register with a username and password of your choice
- Verify you can login and use the flash cards.


Assignment

Practice Test

Practice Test





Practice test available

- Available on Canvas
- Work alone or together in study groups
- Use the forum to compare answers and approaches to questions
- Test #2 will be graded by looking at both your answers to the questions and the work you did on the testing server.

Practice Test Instructions

HONOR CODE:
This is a practice test and you may work with others on it. You are encouraged to compare and discuss answers with your classmates using the forum, study groups or both. However on the real test you must work alone.

INSTRUCTIONS:
Test system: sun-hwa-p2.cis.cabrillo.edu (port 22)

This test should be completed using the sun-hwa-p2 system only. Because this system is on a private network, log into Opus-II first, then ssh into sun-hwa-p2. The practice test and the sun-hwa-p2 system will **not be available** after the real test starts.

Grading will be based on your answers AND that you correctly implemented the "DO THIS FIRST" portion of the question.

If you get stuck on a practice test question you can ask your classmates for help on the forum. If you get stuck on the real test and can't proceed you can ask the instructor for help and forfeit the point. For the real test the instructor will be available during class and available by email later in the evening from 8:00-10:00PM.

Please KEEP YOUR ANSWERS TO A SINGLE LINE ONLY !!

This is a practice test and unlike the real test you can take it as many times as you want. To prepare for the real test keep taking this practice test over and over again till you can answer each question in under 30 seconds.

P = 5 minutes before class ends *(noon or 4pm)*
T = when real test starts *(11am or 3pm)*
T-30 = 30 minutes before real test starts *(10:30am or 2:30pm)*

Reminder to instructor:

On Canvas
[ ] Schedule Practice Test from **P** till **T-30**
[ ] Publish Practice Test
[ ] Moderate any accommodations
[ ] Remove password on practice test
[ ] Update test Q21 for number of accounts

On Practice Test system
[ ] echo "/root/unlock-cis90; passwd -l cis90; rm /etc/nologin" | at **P**
[ ] echo "/root/lock-cis90; cp /etc/nologin.bak /etc/nologin" | at **T-30**


Wrap up

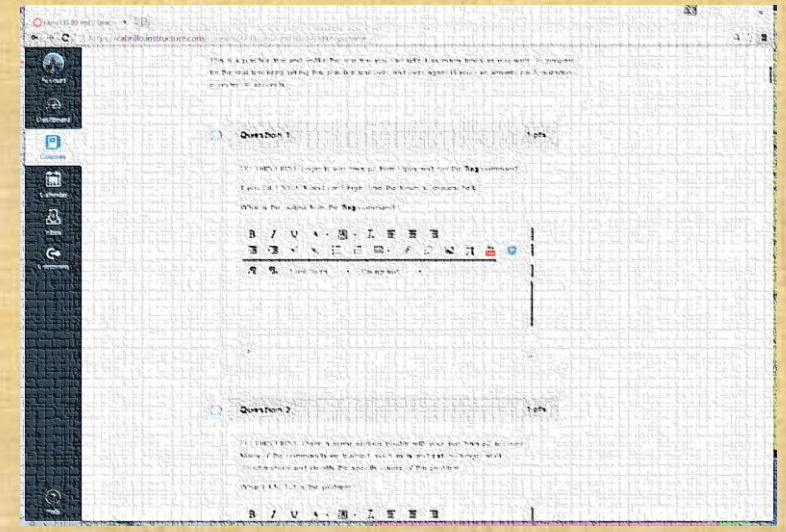
Next Class

No Quiz

Cumulative Test (30 points) with focus on Lessons 6-9:

- Recommended preparation:
  - Work the practice test!
  - Restore your directory with ./restore and work the practice test again!
  - Repeat step above till you can answer all questions in 30 seconds
  - Make a personal reference "crib sheet" document
  - Collaborate with others on the forum to compare answers
  - Review Lessons 6-9 slides and Labs 5-7
  - Try doing some or all of Lab X2 (pathnames)
  - Practice with flash cards
  - Scan previous Lessons so you know where to find things if needed

If time permits



Work the practice test till the end of class.

- Collaborate!
- Ask questions!
- Arrange study groups!

Backup