



Rich's lesson module checklist

Last updated 4/5/2018

	Zoom recording named and published for previous lesson			
	Slides posted Print out agenda slide and annotate page numbers			
	1 st minute quiz Flash cards Calendar page updated			
	Update test Q21 for number of accounts 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			
_ _ _	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



Shell commands

Permissions

Secure logins

Processes

CIS 90 Introduction to **UNIX/Linux**

Navigate file tree

Scheduling tasks

The Command Line

Files and directories

Mail

vi editor

Environment variables

Shell scripting

Filters

Pipes

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.







Jim Griffin

- Created this Linux course
- Created Opus and the CIS VLab
- Jim's site: 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

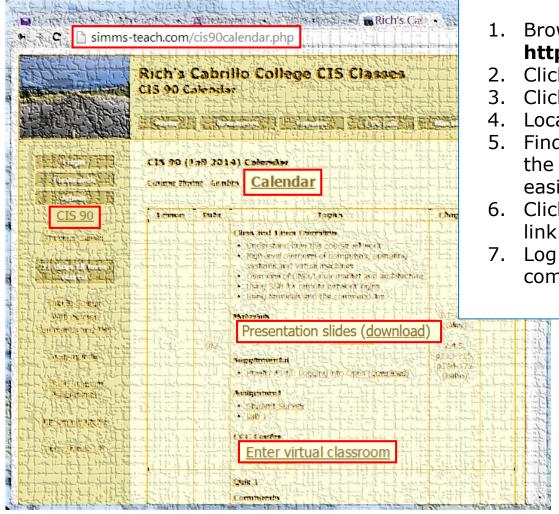
And thanks to:

 John Govsky for many teaching best practices: e.g. the First Minute quizzes, the online forum, and the point grading system (http://teacherjohn.com/)





Student checklist - Before class starts



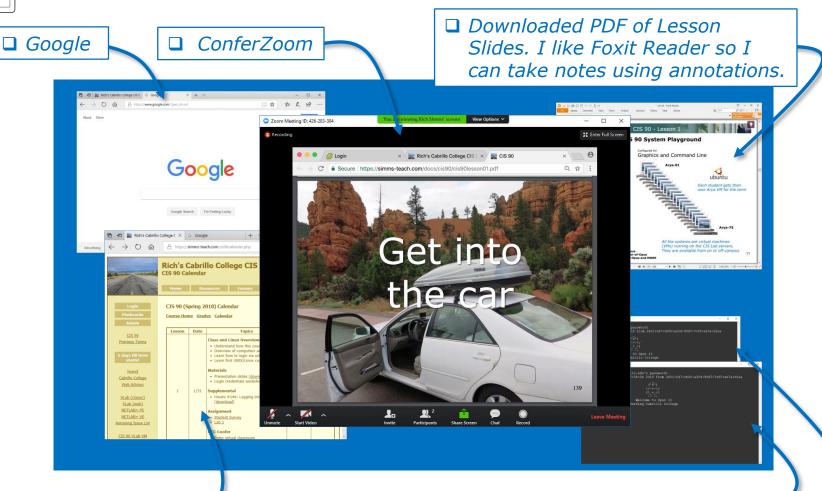
1. Browse to: http://simms-teach.com

- 2. Click the CIS 90 link.
- Click the <u>Calendar</u> link.
- 4. Locate today's lesson.
- Find the Presentation slides for the lesson and <u>download</u> 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



☐ CIS 90 website Calendar page □ One or more login sessions to Opus-II



Start

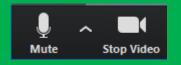




Start Recording

Audio Check





Start Recording

Audio & video Check



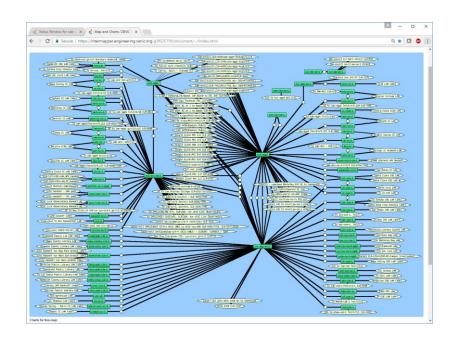
CIS 90 - Lesson 9



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

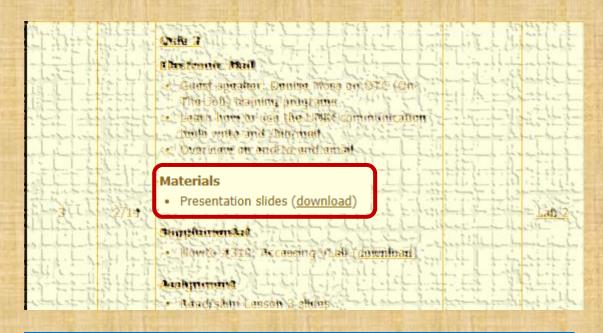
```
('v')
\/-=-\/
(\_=_/)
~~ ~~

Welcome to Opus II
Serving Cabrillo College
```

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



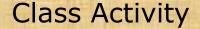
Class Activity

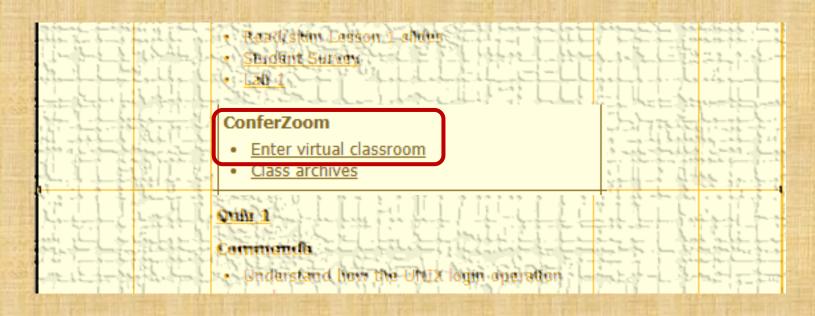


https://simms-teach.com/cis90calendar.php

If you haven't already, download the lesson slides







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

If you haven't already, join ConferZoom classroom









Questions?

Lesson material?

Labs? Tests?

How this course works?

. Graded work in the street ories home directories.

. Answers in cis90 answers home cis90 home

Who questions much, shall learn much, and retain much.

- Francis Bacon

If you don't ask, you don't get.

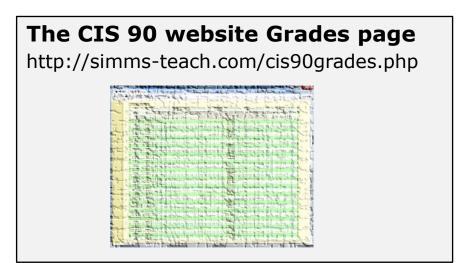
- Mahatma Gandhi

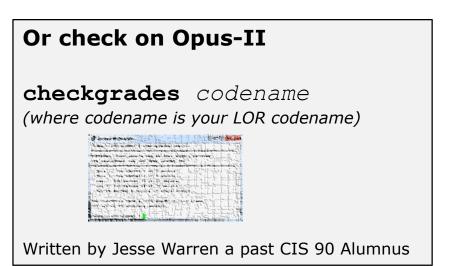
Chinese Proverb 他問一個問題, 五分鐘是個傻子, 他不問一個問題仍然是一個傻瓜永遠。

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





Send me your survey to get your LOR code name.

Percentage	Total Points	Letter Grade	Pass/No Pass
90% or higher	504 or higher	Α	Pass
80% to 89.9%	448 to 503	В	Pass
70% to 79.9%	392 to 447	С	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:
6 labs:
1 test:
2 forum quarters:

18 points
180 points
30 points
40 points
268 points





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.

- Log into your Arya VM as the cis90 user. Make sure it's your VM and not someone else's.
- Install the latest updates: sudo apt-get update sudo apt-get upgrade
- 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 6 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

The parts content review - The first person to email the instructor pointing out an
error or type on this welsals will get one point of extra credit for each unique error.
The email must specify the specify document or well page, proponit the location of the
error, and specify what the correction should be. Duplicate errors count as a single
upoint. This does not apply to pre-published material than has been optoided but not
wet presented in class. (Up to 20 points total)



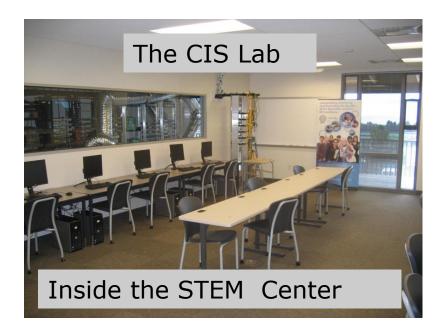
Getting Help When Stuck on a Lab Assignment

- Google the topic/error message.
- Search the Lesson Slides (they are PDFs) for a relevant example on how to do something.
- Post a question on the forum. Explain what you are trying to do and what you have tried so far.
- Talk to a STEM center tutor/assistant.
- Come see me during my office or lab hours. I will be in the CTC (room 1403) every Wednesday afternoon from 3-5:30.
- 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!

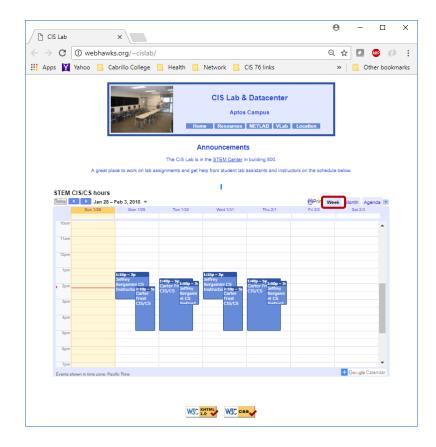


Help Available in the CIS Lab

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

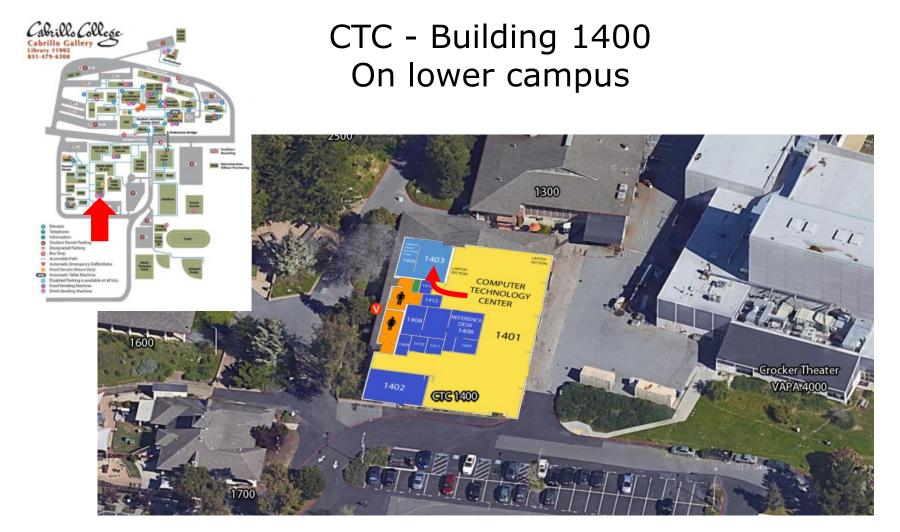












I will be in the CTC (room 1403) every Wednesday afternoon from 3-5:30





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 ...

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

Email: risimms@cabrillo.edu







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





Donate by answering seven questions on an online CTE survey!





Perkins/VTEA Survey Please complete the survey by Friday, April 6th



https://opus-ii.cis.cabrillo.edu/forum/viewtopic.php?f=6&t=349

This is an important source of funding for Cabrillo College.

Send me an email by Friday April 6th stating you completed this Perkins/VTEA survey for three points extra credit!

		hnical Information rs to these questions will help qualify Cabrillo College for Perkins/VTEA grant funds.
١	Are you curr	rently receiving benefits from:
١	Yes	TANF/CALWORKS
١	No	
١	Yes	SSI (Supplemental Security Income)
١	No	
١	Yes	GA (General Assistance)
١	No	
١	Yes	Does your income qualify you for a fee waiver?
١	No	
١	Yes	Are you a single parent with custody of one or more minor children?
١	No	
١	Yes	Are you a displaced homemaker attending Cabrillo to develop job skills?
١	No	
١	Yes	Have you moved in the preceding 36 months to obtain, or to accompany parents or spouses to obtain,
١	No	temporary or seasonal employment in agriculture, dairy, or fishing?



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:

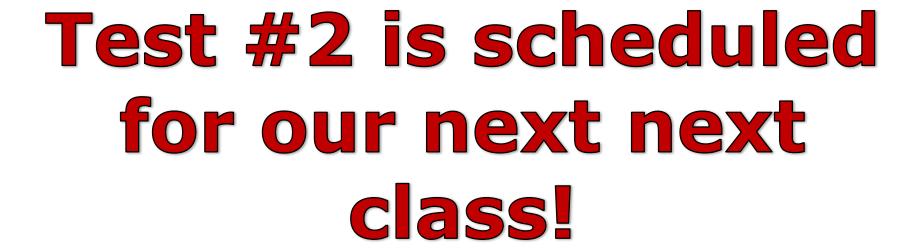
Test #2 is scheduled for our next class!



Test #2 will happen during our next class!

Practice test available after class.





Practice test available after class.





Test #2

- Test #2 is scheduled for our next class! Same format as before. The test will start during the last hour of class. If you work you can take it later in the day as long as it is completed by 11:59PM.
- 2. Practice Test #2 is available after class on Canvas!
- 3. Work the Practice Test BEFORE the real test begins.
- 4. The Practice Test and Practice Test server will be available until about 30 minutes before the real test starts.

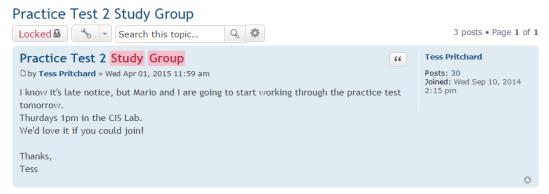


How to pass Test #2 with flying colors

- 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.
- 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!
- Create a custom crib sheet of commands and key concepts covered in the course.
- 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!

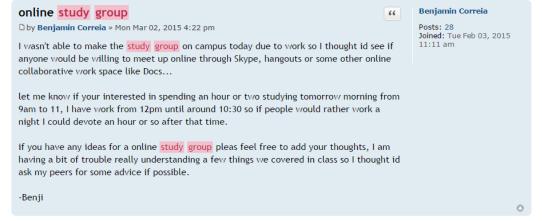


Use the forum to arrange study groups



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.

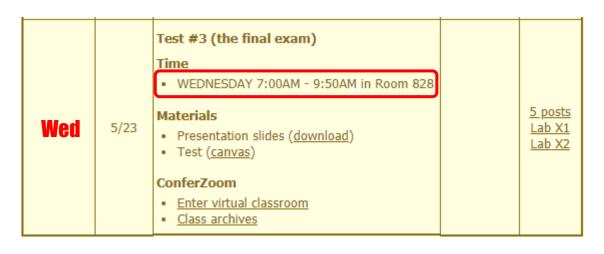


Subscribed looks like this.



Heads up on Final Exam

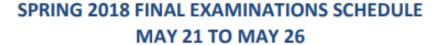
Test #3 (final exam) is Wednesday May 23rd 7-9:50AM



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

- All students will take the test at the <u>same</u> <u>time</u>. The test must be completed by 9:50_{AM}.
- 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)





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 21
9:00 am and 10:15 am, MW/Daily	7:00 am-9:50 am	Wednesday, May 23

CIS 90 Introduction to UNIX/Linux

Provides a technical overview of the UNIX/Linux operating system, including handson 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
&	Arr.	Arr.	R.Simms	OL

Section 1-(102385) 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
Socti	on 2 (102286	Nie a Hybrid ONLINE	course Mosts weekly	throughout the

Section 2-(102386) 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.





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



The Urban Penguin

Welcome to this self study video series of tutorials. These videos can be used in preparing you for the LPI.(Linux Professional Institute). Linux Essentials Certification. These materials are meant as a stand-alone learning solution in readiness for your exam and are targeted towards anyone who is aiming for the certification or just wants to know more about what Linux is and what it can offer. The Urban Penguin is an Approved LPI Training Partner and we provide both free training via these videos and, if you prefer to work direct with the penguin, there we can offer online training at a reasonable cost.			
Objective	Description	Click to Access	
Intro	What is LPI Linux Essentials	Click to Access	
1.1	Linux evolution and popular operating systems	Click to Access	
1.2	Major Open Source applications	Click to Access	
1.3	Understanding Open Source Software and licensing	Click to Access	
1.4	ICT skills and working with Linux	Click to Access	
2.1	Command line basics	Click to Access	
2.2	Using the command line to get help	Click to Access	
2.3	Using directories and listing files	Click to Access	
2.4	Creating, moving and deleting	Click to Access	
3.1	Archiving files from the command line	Click to Access	
3.2	Searching and extracting data from files	Click to Access	
3.3	Turning commands into a script	Click to Access	
4.1	Choosing an operating system	Click to Access	
4.2	Understanding computer hardware	Click to Access	
4.3	Where data is stored	Click to Access	
4.4	Your computer on the network	Click to Access	
5.1	Basic security and user types	Click to Access	
5.2	Creating users and groups	Click to Access	
5.3	Manage file permissions and ownership	Click to Access	
5.4	Special directories and files	Click to Access	

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.





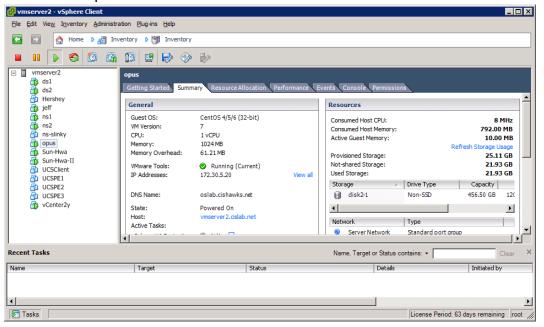


Our Opus-II server on campus

Dell R610 Server



VMware vSphere Client





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







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 See: http://iso.linuxquestions.org/

2)



Get a USB flash drive

Google "boot live linux from usb" for instructions

3) *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



CentOS









YUMI formatted Flash Drive (www.pendrivelinux.com)

Linux Mint



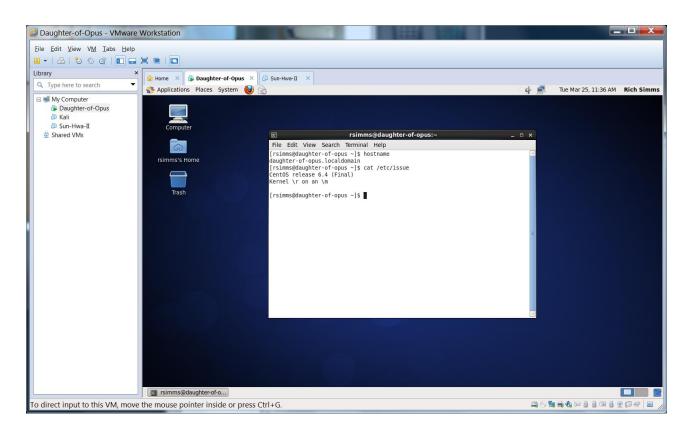


Allows you to use or try out Linux on an existing computer without installing it



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



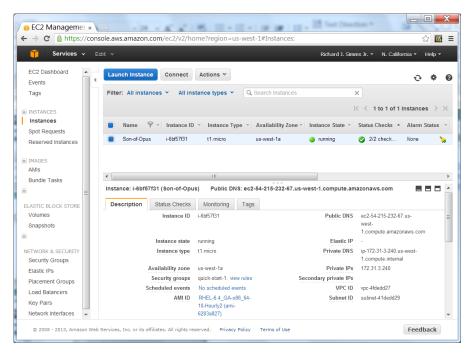
This Daughter-of-Opus is a VM running on my laptop using Oracle VirtualBox (never expires)





Amazon Web Services





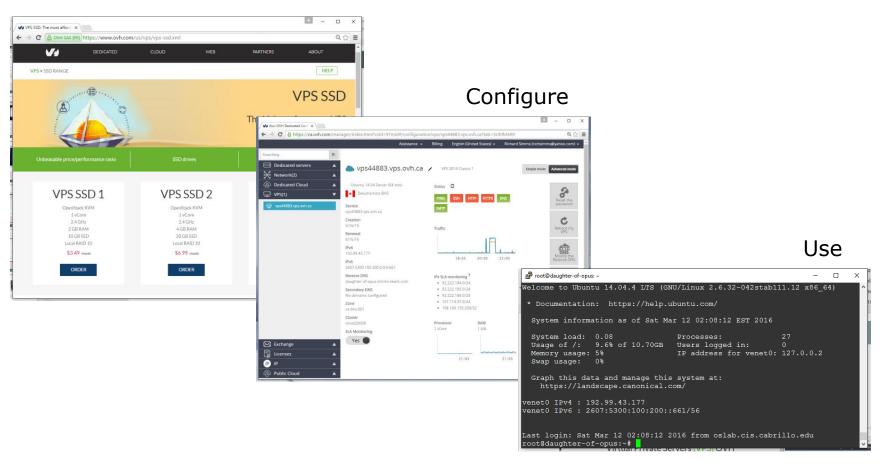


Son-of-Opus is a VM running on Amazon Web Services



OVH.com

Purchase



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/produc t-catalog/servers/proliantservers.filtersfacet_subbrand_url:ProLiant-MicroServer.hits-12.html





SuperMicro

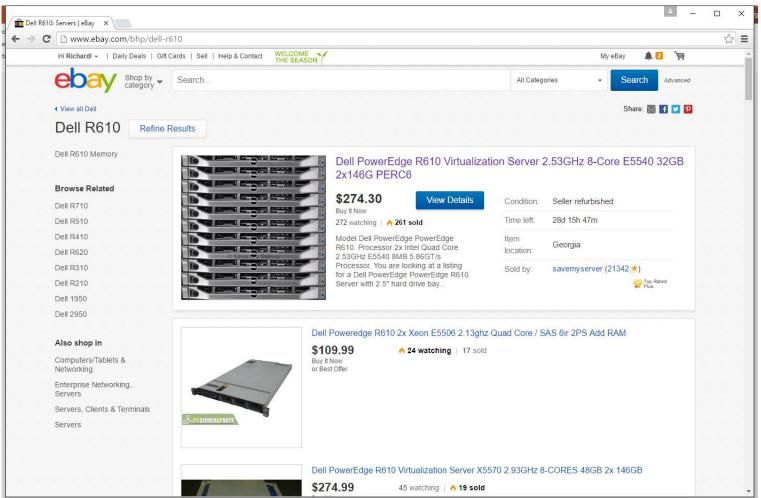


https://tinkertry.com/my-tinkertry-dxeon-d-bundle-2-supermicrosuperserver-bundle-2-of-joy

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

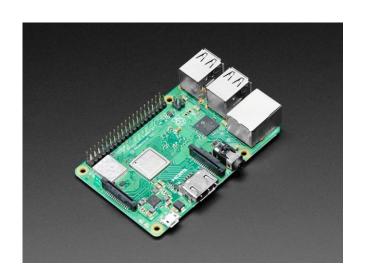


Fantastic Bargains on EBay











Raspberry Pi 3 Model B+

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

Raspberry Pi Zero WH

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



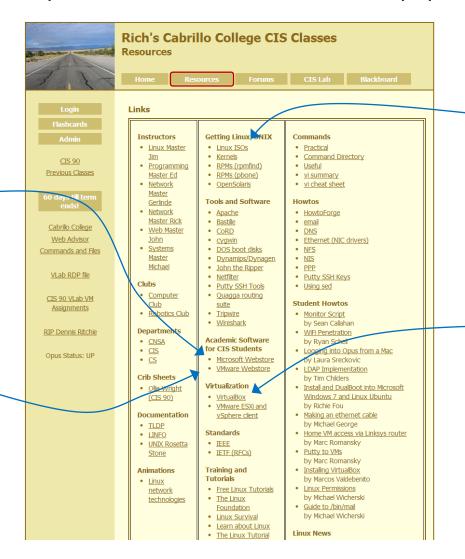


Microsoft Software (Academic)

VMware Software (Academic)



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



Linux Distros (ISOs)



VirtualBox









File Redirection

The 3 standard UNIX file descriptors:

Name	Integer Value
stdin (standard in)	0
stdout (standard out)	1
stderr (standard error)	2

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





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

№ filename

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

X> 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

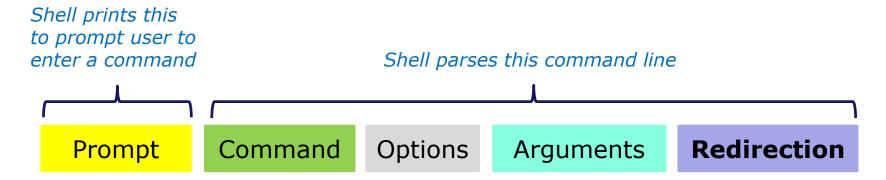
Redirects **stdout**, output will now be appended to *filename*.

2>> filename

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



The redirection is specified on the command line



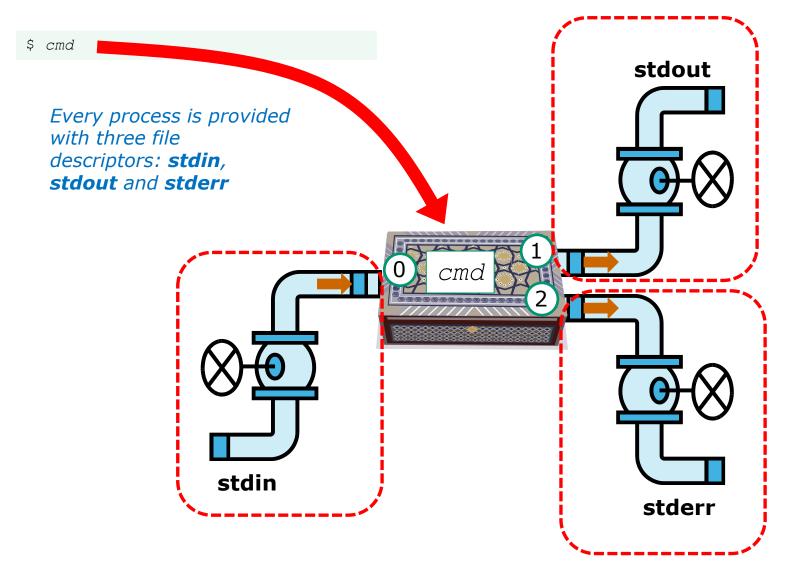
mples Redirection connects stdin, stdout and stderr to non-default devices

Examples

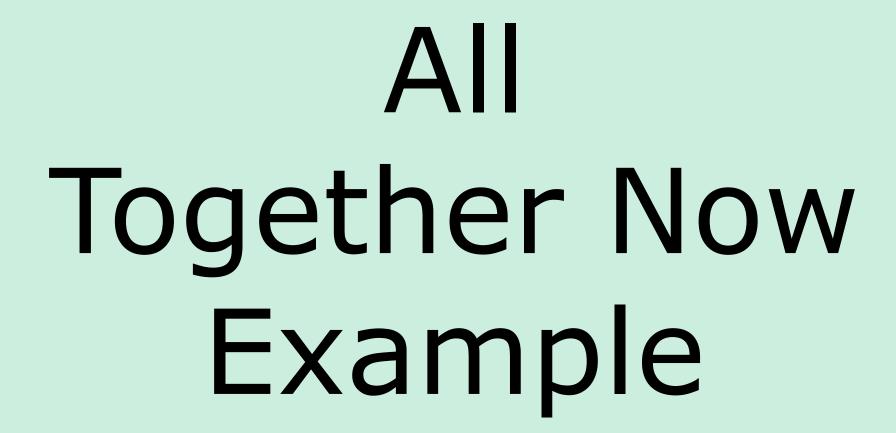
```
/home/cis90/simben $ cat
                    cat
/home/cis90/simben
                         -A letter
/home/cis90/simben
                   $ cat
                            < letter
/home/cis90/simben
                   $ cat
                         -b < letter > out
                         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











Life of the Shell

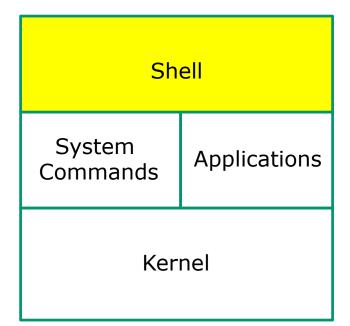












- 1) Prompt
- 2) Parse
- 3) Search
- 4) Execute
- 5) Nap
- 6) Repeat



Example



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

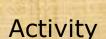
The shell begins by echoing a **prompt** string to your terminal device:

- 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 \$

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.







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

The prompt is defined by your PS1 variable

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

Example



- 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 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









Prompt

- Parse
- 3) Search
- 4) Execute
- 5) Nap
- 6) Repeat

Activity

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 size of your five file: Is -I five
- 3. 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.
- 4. Put the size of your *five* file and the hex value of the newline character in the chat window.
- 5. Optional: Use **man ascii** to check your answer.

Example

1) Prompt



- 2) Parse
- 3) Search
- 4) Execute
- 5) Nap
- 6) Repeat

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.

/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**



Example

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

/usr/local/bin:/usr/bin:/usr/local/sbin:

/usr/sbin:/home/cis90/simben/../bin:
/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.

00

- l) Prompt
- 2) Parse



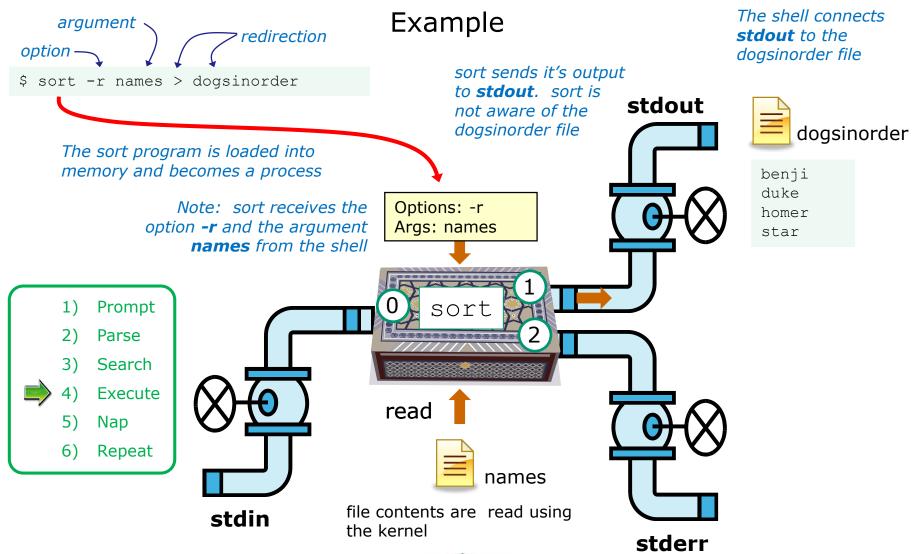
- 3) Search
- 4) Execute
- 5) Nap
- 6) Repeat

Activity

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

- 1. Use **echo \$PATH** to view your path.
- 2. Starting with the first directory on the path look for the sort command:
 - Use Is -li /usr/local/bin | grep sort
 - Use Is -li /usr/bin | grep sort
- 3. Write the inode number of the sort program file in the chat window.





sort opens and reads the names file



Activity

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

\$ sort -r names > dogsinorder

What two text strings parsed by the shell were passed 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
- S) Repeat

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







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.

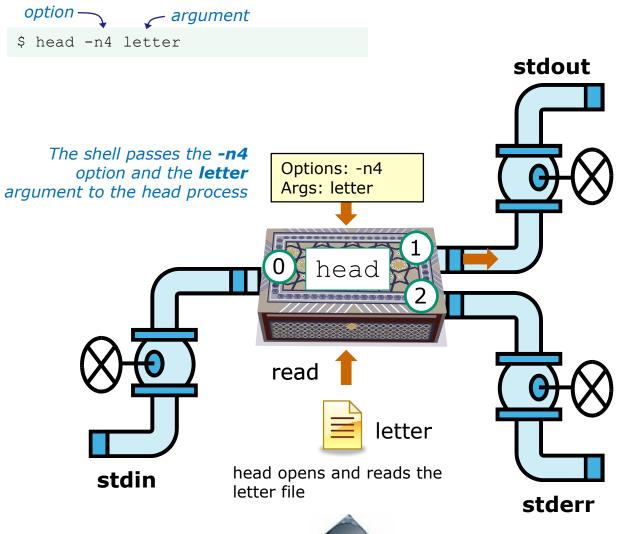
/home/cis90/simben \$ head -n4 < letter
Hello Mother! Hello Father!</pre>

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



CIS 90 - Lesson 9

head -n4 letter



Hello Mother! Hello Father!

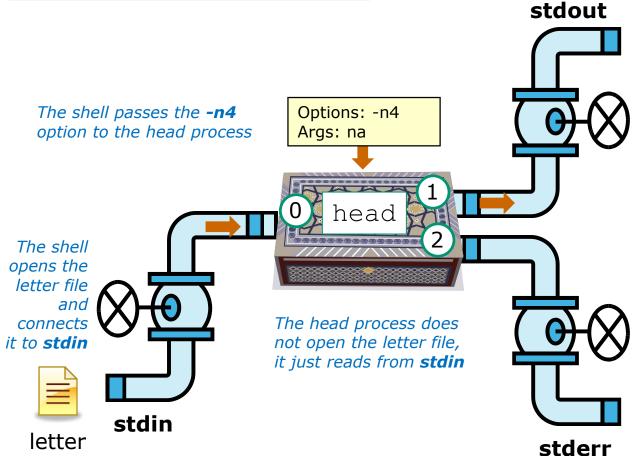
Here I am at Camp Granada. Things are very entertaining, and they say we'll have some fun when it stops raining.







\$ head -n4 < letter</pre>



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.





Instructor: Using ConferZoom annotations



CIS 90 - Lesson 9

ConferZoom Annotations





View Options > Annotate Draw > "/"









Select the straight line drawing tool and connect the like images













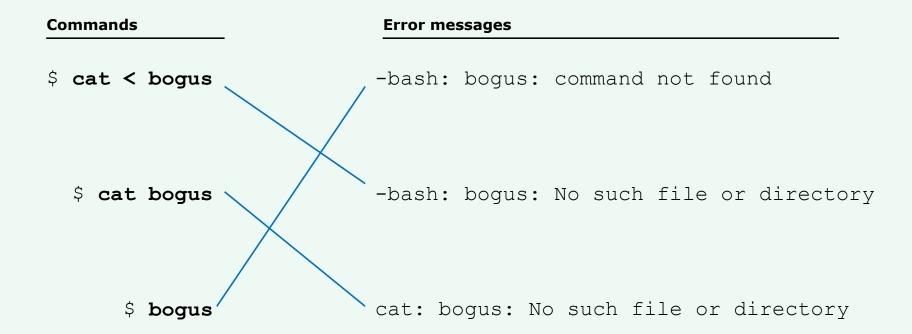


Connect with a straight line the command with the error message

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



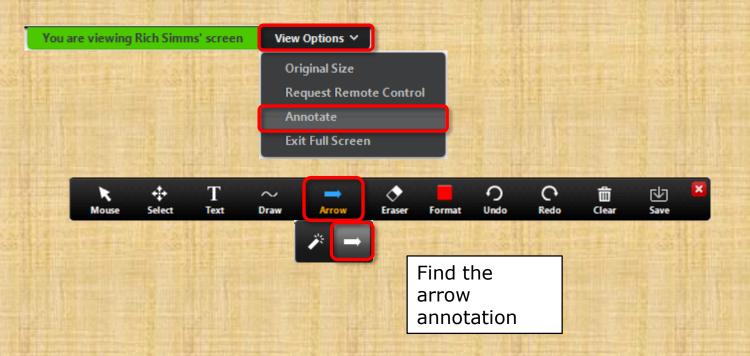
Connect with a straight line the command with the error message





CIS 90 - Lesson 9

ConferZoom Annotations



View Options > Annotate Draw > Spotlight or Arrow > Arrow









l

Point at the number called out by the instructor with your arrow annotation.





Shell Steps

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

1) Prompt

2) Parse

3) Search

4) Execute

5) Nap

6) Repeat





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





Shell Steps

1) Prompt

2) Parse

3) Search

4) Execute

5) Nap

6) Repeat

Given: There is no file named bogus

[rsimms@oslab ~]\$ cat < bogus
-bash: bogus: No such file or directory</pre>

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

Given: There is no file named bogus

[rsimms@oslab ~]\$ bogus1 < bogus2
-bash: bogus2: No such file or directory</pre>

Point to the shell step where the error message was generated



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</pre>
```



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





It's descriptor clobbering time!

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

```
stdout (next free byte)
         stderr (next free byte)
           stdout (next free byte)
2+2
       stderr (next free byte)
                                  The <newline> character is represented by a "."
          stdout (next free byte)
7/0
      Runtime error (func=(main), adr=5): Divide by zero.
                                                                              stderr (next free byte)
             c stdout (next free byte)
3 + 3
       Ru6.ime error
                           (func=(main)
                                                                               stderr (next free byte)
       /home/cis90/simben $ cat calculations
        Ru6
       ime error (func=(main), adr=5): Divide by zero
```





It's descriptor collaboration time!

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

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

```
/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





(input/output)

programming examples







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 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





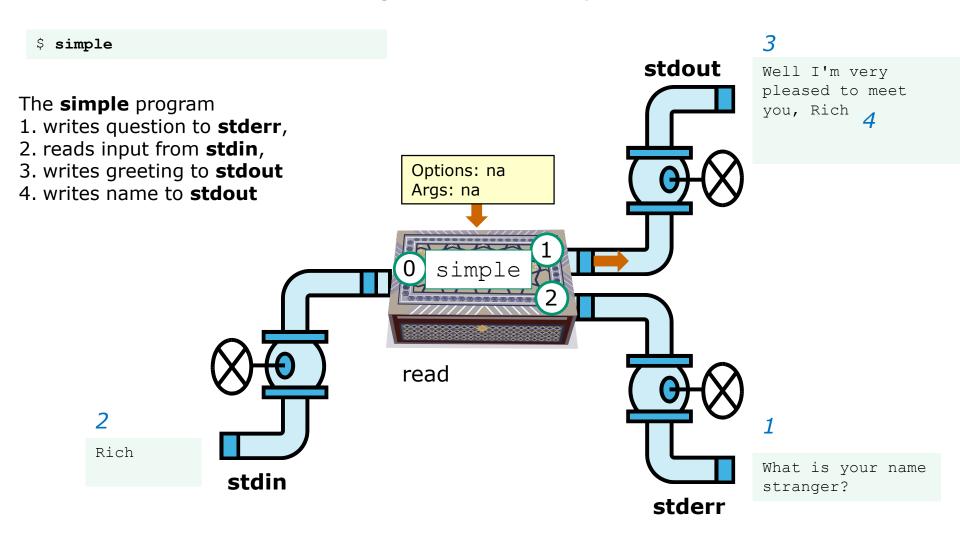
C Program I/O example Execute the program

/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







C Program I/O example

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

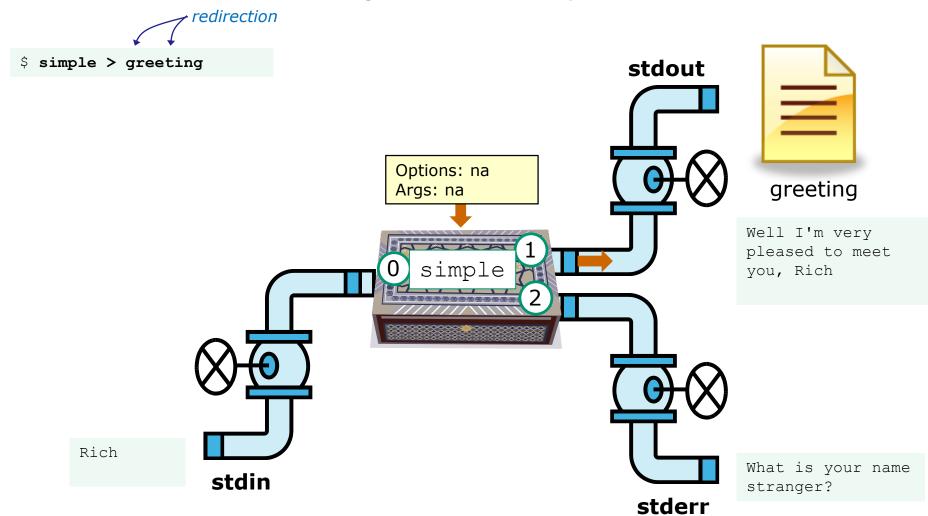
In this example, output has been redirected to a file named myfile.

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

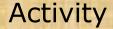
The simple program has no special knowledge (coding instructions) for a file named myfile. It just writes to **stdout** and that output will go to wherever **stdout** had been directed.



C Program I/O example







- Change to your bin directory
 cd ~/bin
- 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





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; ←
                                - Write question to stderr
    string buffer;
                              Read name from stdin
    cin >> buffer; 
    string greeting = "Well I'm very pleased to meet you, ";
    cout << greeting << buffer << endl;</pre>
    return 0;
                                          Write greeting and name to stdout
```





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

- Change to your bin directory
 cd ~/bin
- 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**





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

- Change to your bin directory
 cd ~/bin
- 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**

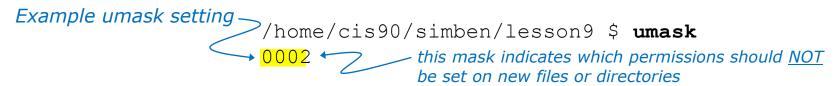




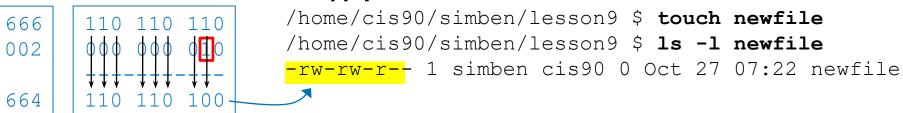
(review)



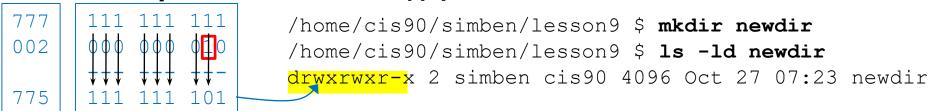




New file - start with 666 and apply mask



New directory - start with 777 and apply mask



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
                                   Example umask setting
/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 simben 90 cis 90 0 Mar 24 17:50 myfile
-rw--w---. 1 simben90 cis90 0 Mar 24 17:51 myfile.bak
    622
                          Copied file - start with original
    057
                          file's permissions and apply the
                          mask
    620
```

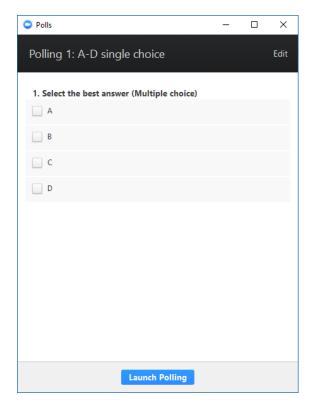
Remember, for new files resulting from copying, instead of using the **default permissions** (666 for file and 777 for directory), use the **original file permissions** as the starting point for the mask to be applied to.





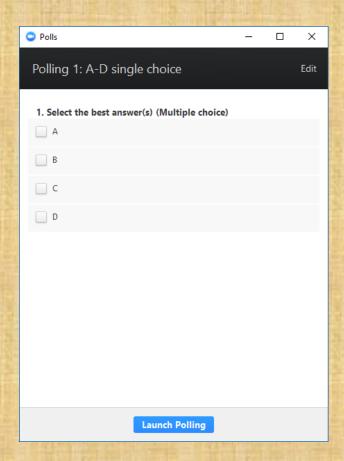








Activity



Which pizza is the best?

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





I want to change the permissions on an existing file

Which command does this?

- A) stat
- B) Is -I
- C) chmod
- D) umask







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

Which command does this?

- A) stat
- B) Is -I
- C) chmod
- D) umask







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) Is -I
- C) chmod
- D) umask







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

Which command does this?

- A) stat
- B) Is -I
- C) chmod
- D) umask











Task

Record the last times Homer Miller logged in on a Sunday to a file named *mylog* AND count them

grep Homer /etc/passwd

```
last | grep milhom90 | grep "Sun" | tee mylog cat mylog last | grep milhom90 | grep "Sun" | tee mylog cat mylog last | grep milhom90 | grep "Sun" | tee mylog 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.





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 5 -d ":" | 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 | 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

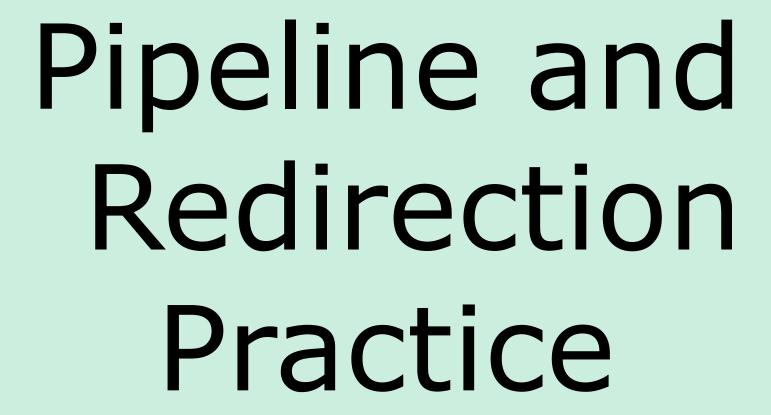
Task

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.







bc command with no redirection or piping

```
/home/cis90/simben $ bc
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free
Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
2+2
4
4/0
Runtime error (func=(main), adr=5): Divide by zero
quit
/home/cis90/simben $
```



Piping output to bc command

```
/home/cis90/simben $ echo 2+2 | bc
4
/home/cis90/simben $ echo 4/0 | bc
Runtime error (func=(main), adr=5): Divide by zero
```



Redirecting stdin of bc command

Setup:

```
/home/cis90/simben $ echo 2+2 > datafile
/home/cis90/simben $ echo 4/0 >> datafile
/home/cis90/simben $ cat datafile
2+2
4/0
```

```
/home/cis90/simben $ bc < datafile
4
Runtime error (func=(main), adr=5): Divide by zero</pre>
```



Piping output to bc command

Setup:

```
/home/cis90/simben $ echo 2+2 > datafile
/home/cis90/simben $ echo 4/0 >> datafile
/home/cis90/simben $ cat datafile
2+2
4/0
```

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



Redirecting stdin, stdout and stderr of bc command

Setup: /home/cis90/simben \$ echo 2+2 > datafile /home/cis90/simben \$ echo 4/0 >> datafile

/home/cis90/simben \$ cat datafile

2+2

4/0

```
/home/cis90/simben $ bc < datafile > results 2> errors
/home/cis90/simben $ cat results
4
/home/cis90/simben $ cat errors
Runtime error (func=(main), adr=5): Divide by zero
```



Piping stdout and redirecting stdin, stderr of bc command

Setup:

```
/home/cis90/simben $ echo 2+2 > datafile
/home/cis90/simben $ echo 4/0 >> datafile
/home/cis90/simben $ cat datafile
2+2
4/0
```

```
/home/cis90/simben $ bc < datafile 2> errors | mail -s "Example" simben90 /home/cis90/simben $ cat errors
Runtime error (func=(main), adr=5): Divide by zero
```





Activity

Setup:

```
/home/cis90/simben $ echo 2+2 > datafile
/home/cis90/simben $ echo 4/0 >> datafile
/home/cis90/simben $ cat datafile
2+2
4/0
```

Example:

```
/home/cis90/simben $ bc < datafile 2> errors | mail -s "Example" $LOGNAME /home/cis90/simben $ cat errors

Runtime error (func=(main), adr=5): Divide by zero
```

Past the email you receive into the chat window







The wc command is a filter.

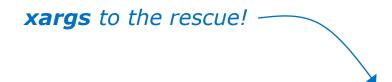
```
/home/cis90/simben $ head -n2 poems/Anon/nursery
Jack and Jill went up the hill
to fetch a pail of water.
/home/cis90/simben $ head -n2 poems/Anon/nursery | wc -l
2
/home/cis90/simben $
```

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



/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 $

/home/cis90/simben $

/home/cis90/simben $
```

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



Another example





The **Is** 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!



```
/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 **Is** 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 **Is -Id** command

xargs to the rescue again!



```
/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.









How to locate files:

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

To locate by user, type, group, etc1. ... use **find**

To locate by contents² within specific files ... use grep

To locate by contents² 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)



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.





Task

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

find /???? -user ?????? 2> /dev/??? | ?? -1

Write your answer in the chat window



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.pv /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.





Task

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

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

Write your answer in the chat window





Task

Find an account for simben 90 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



Finding Things

Task

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

grep -r "Torvalds" /usr/src

```
| Usurforchemete/2, 6, 32-220, 23, 1.elis.i686/archy/s9x9/mkeffile: # Copyright (C) 1994 by Linus Torvadds | Usurforchemete/2, 6, 32-220, 23, 1.elis.i686/archy/s9x9/mkeffile: # Copyright (C) 1994 by Linus Torvadds | Usurforchemete/2, 6, 32-220, 23, 1.elis.i686/archy/s9x9/mkeffile: # Copyright (C) 1994 by Linus Torvadds | Usurforchemete/2, 6, 32-220, 23, 1.elis.i686/archy/s9x9/mkeffile: # Copyright (C) 1994 by Linus Torvadds | Usurforchemete/2, 6, 32-220, 23, 1.elis.i686/archy/s8p/inched/sam/1373, https://doi.org/10.1001/mkeffile: # Copyright (C) 1994 by Linus Torvadds | Usurforchemete/2, 6, 32-220, 31, elis.i686/archy/s8p/inched/sam/1373, https://doi.org/10.1001/mkeffile: # Copyright (C) 1994 by Linus Torvadds | Usurforchemete/2, 6, 32-220, 31, elis.i686/archy/s8p/inched/sam/1482, https://doi.org/10.1001/mkeffile: # Copyright (C) 1995 Linus Torvadds | Usurforchemete/2, 6, 32-220, 31, elis.i686/archy/s8p/inched/sam/1482, https://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile://doi.org/10.1001/mkeffile
```

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











Egg Hunt

Instructor: sudo /home/rsimms/cis90/basket/hide-the-eggs

A number of colored eggs 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. Who is the owner and group for this egg file? The rest are scattered in the various subdirectories you own.
- 2. Make a new directory named *basket* in your home directory and see how many egg files you can move into it.
- 3. Put a Green Check in ConferZoom next to your name when you have collected 3 eggs, put a "thumbs up" if you collect all 17.









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 Online Teams









Everyone needs to be on ConferZoom today

Instructor:

PS1="[\u@\h \W] \t \\$ "

/home/rsimms/scripts/teams (use teams alias)





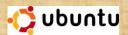


CIS 90 - Lesson 9

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 -qt 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

	Rich's Cabrillo College CIS Classes Login Page
1	Home Resources Forums CIS Lab CTC
Login Flashcards Admin	Please Login You need to login first Username: Password:
CIS 90 CIS 192 Previous Classes	Login
87 days till term ends!	New users click <u>here</u>
<u>Cabrillo College</u> <u>Static IPs</u>	
Metal Sitemap W3C 1.0 W3C css Credits Earth	

Register if this is the first time using Flashcards

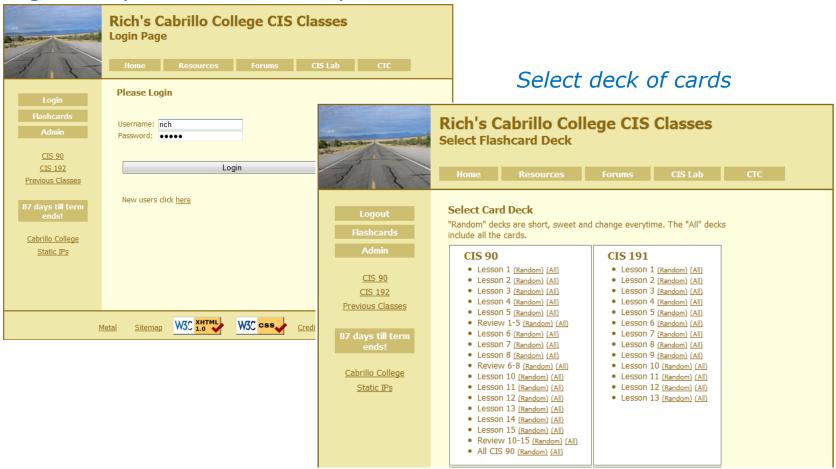


Register and choose a username and password of your choice 169



Logging in and using Flashcards

Login with your username and password







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

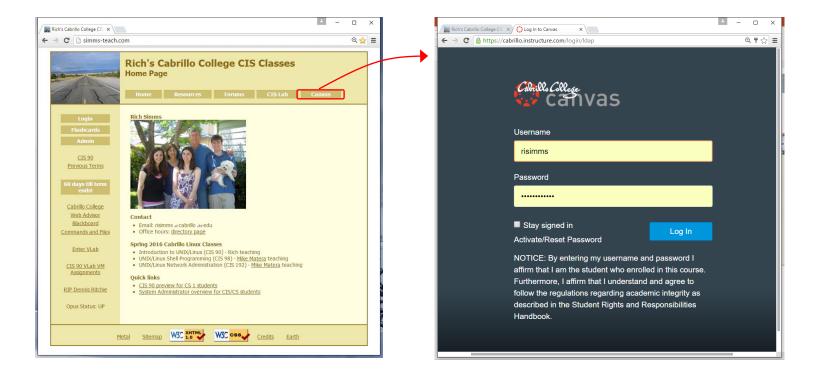








Practice Test



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.





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

CIS 90 - Lesson 9





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 primary system

- echo "/root/unlock-cis90; rm /etc/nologin" | at P
- echo "/root/lock-cis90; cp /etc/nologin.bak /etc/nologin" | at T-30



