Lesson Module Checklist

- Slides
- WB

- Flash cards
- Page numbers
- 1st minute quiz
- Web Calendar summary
- Web book pages
- Commands

- Opus - hide script tested
- Practice test uploaded
- Sun-Hwa - trouble made and test script run

- 9V backup battery for microphone
- Backup slides, CCC info, handouts on flash drive

Student checklist

1) Browse to the CIS 90 website Calendar page
   - http://simms-teach.com
   - Click CIS 90 link on left panel
   - Click Calendar link near top of content area
   - Locate today's lesson on the Calendar

2) Download the presentation slides for today's lesson for easier viewing

3) Click Enter virtual classroom to join CCC Confer session

4) Connect to Opus using Putty or ssh command

Introductions and Credits

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/)

CIS 90 - Lesson 9

Instructor: **Rich Simms**
Dial-in: **888-450-4821**
Passcode: **761867**


Buzz


Carlos


Elijah


Emily

Enrique C.

Enrique R.


Jon M.


Jon W.

Jordan


Joseph


JJ


Kiernan


Maria


Mathew


Mike C.


Michael F.

Mike M.

Nick L.

Patrick


Rebecca

Ricardo


Robert

Ruth


Steve


Tess

Tim


Troy

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

Quiz

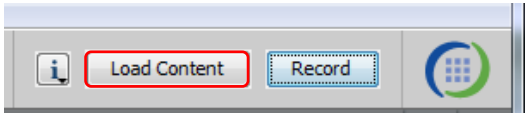
Please answer these questions **in the order** shown:

See electronic white board

email answers to: risimms@cabrillo.edu

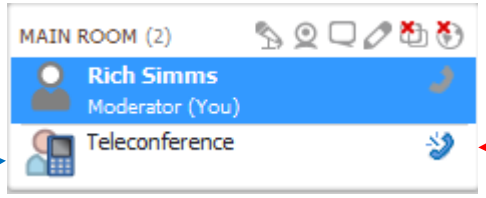
(answers must be emailed within the first few minutes of class for credit)

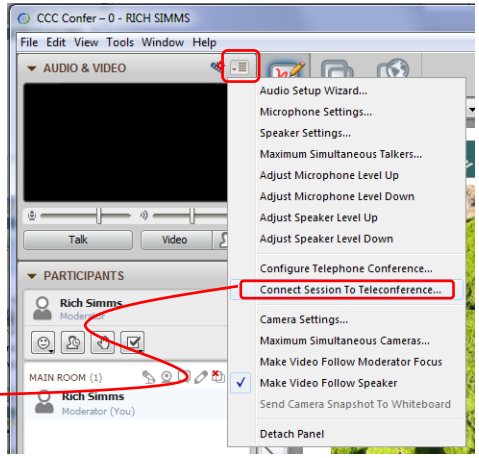
[ ] Preload White Board with *cis*lesson??*-WB*



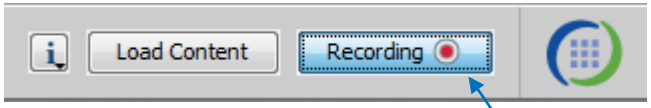
[ ] Connect session to Teleconference

Session now connected to teleconference





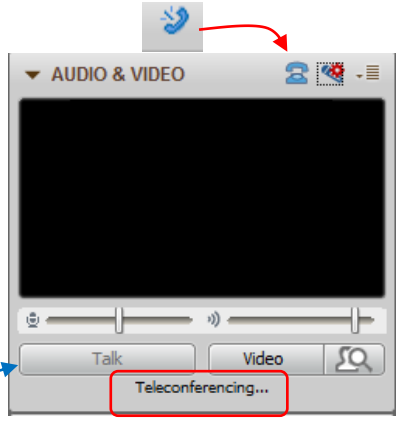
[ ] Is recording on?



Red dot means recording

[ ] Use teleconferencing, not mic

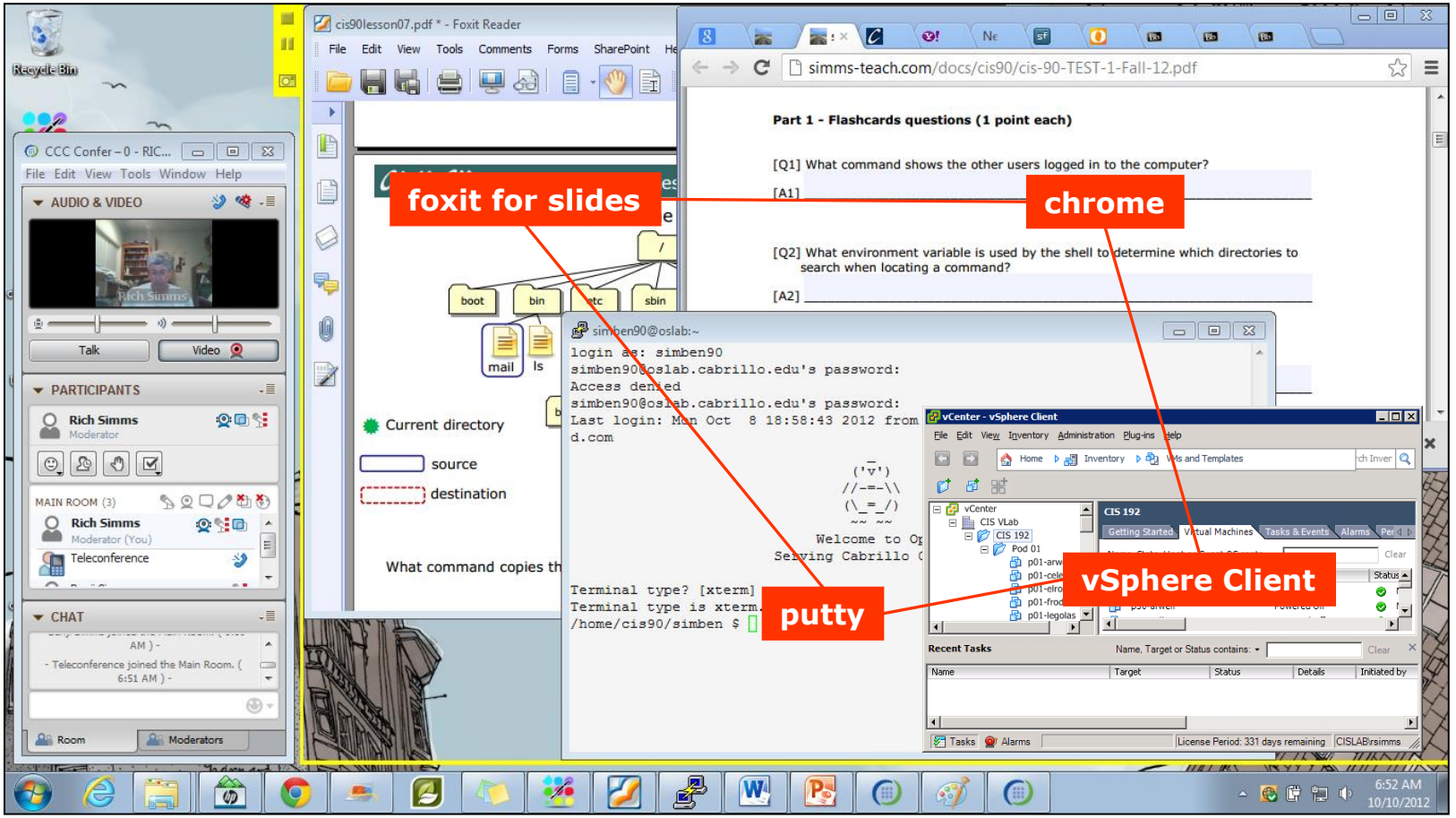
Should be greyed out



Keep wireless mic transmitter away from cell phone and podium if excess static occurs

CCC Confer

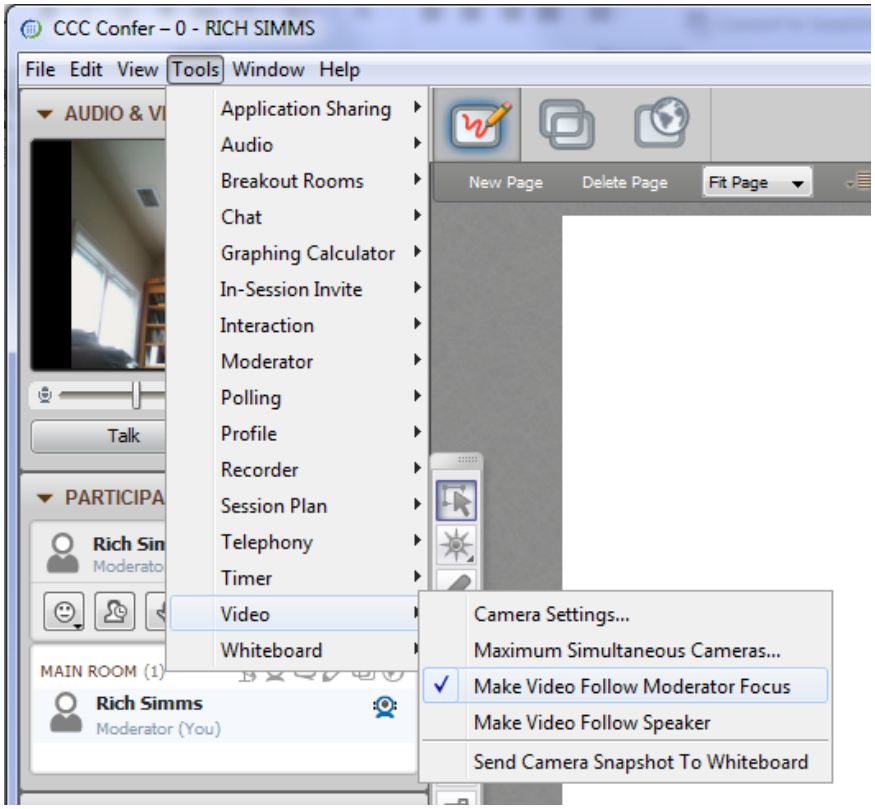
[ ] layout and share apps



[ ] Video (webcam) optional

[ ] Follow moderator

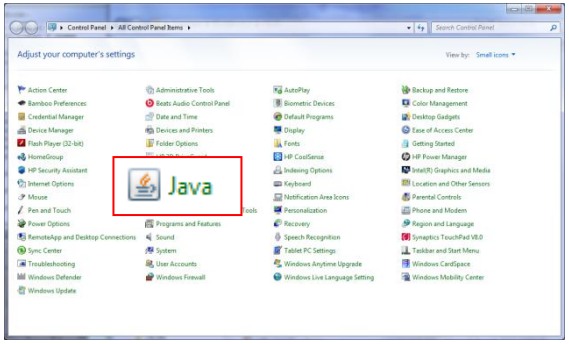
[ ] Double-click on postages stamps



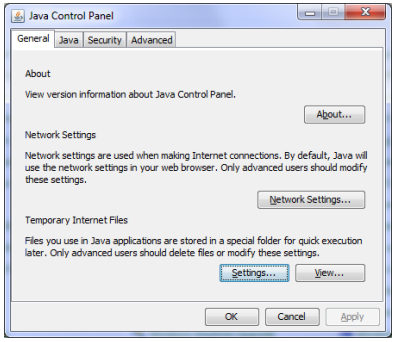
Universal Fix for CCC Confer:

1) Shrink (500 MB) and delete Java cache
2) Uninstall and reinstall latest Java runtime

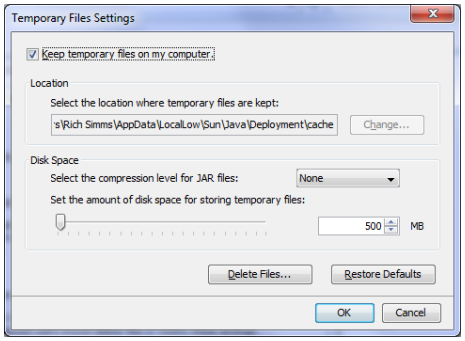
Control Panel (small icons)



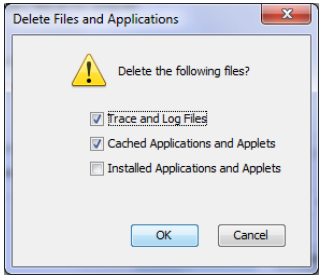
General Tab > Settings…



500MB cache size



Delete these



Google Java download

Review

Objectives	Agenda
• Get ready for the next test • Practice skills • Introduction to processes	• Quiz • Questions • More on I/O • Shell six steps • Subtle I/O • 2>&1 • C program I/O • More on umask • Pipeline practice • Housekeeping • Wireless Penetration (Ryan) • Test Review • Wrap up • Practice test workshop

Questions

Questions?

Lesson material?

Labs? Tests?

How this course works?

- Graded work 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.*

Lab 6
Post Mortem

Lab 6 Results

(steps where points were taken off)

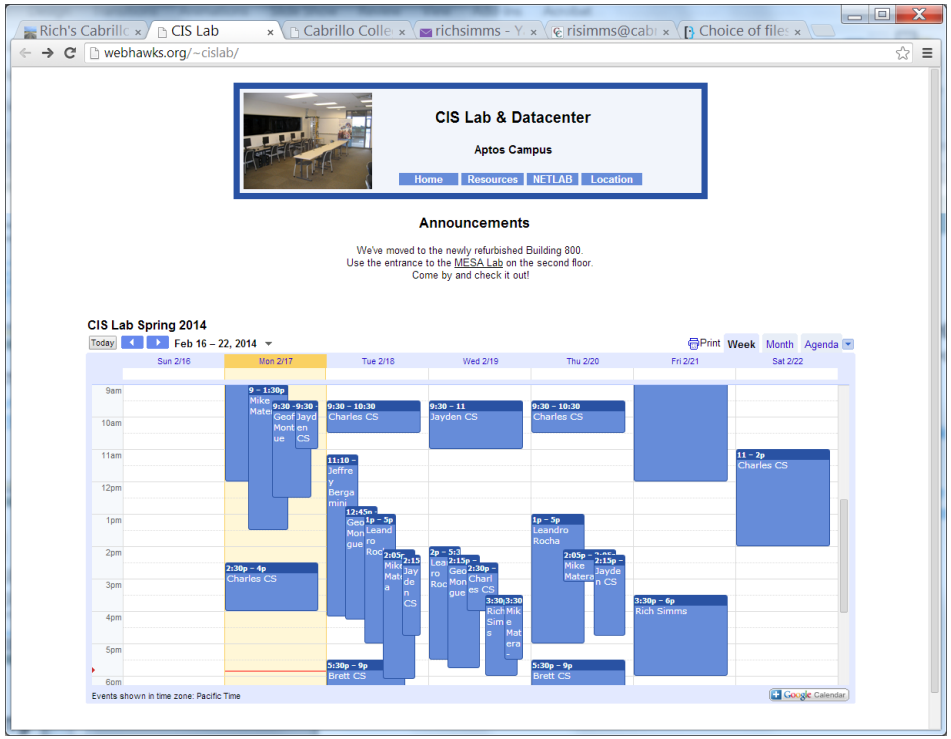
For more on Steps 9-10 see the Backup Slides in Lesson 8 (module titled Lab 6 Tips)

Step 01 -	
Step 02 -	
Step 03 -	
Step 04 -	
Step 05 - x	
Step 06 -	
Step 07 -	
Step 08 -	
Step 09 - xxxxxxx	Set the permissions of your poems directory and …
Step 10 - xxxxxx	Set all ordinary files under the poems …
Step 11 - xxxx	Change the permissions of your bin …
Step 12 - xxx	
Step 13 - x	
Step 14 - xxx	
Step 15 - xxxxx	Make all ordinary files under class/labs and …
Step 16 - xx	
Step 17 - x	
Step 18 -	
Step 19 - x	
Step 20 -	
Step 21 - x	
Step 22 -	
Step 23 - xxxxx	Try setting the umask to 777 …

Correct Lab 6 submittal available in /home/cis90/answers directory on Opus

CIS Lab Schedule

http://webhawks.org/~cislab/



Not submitting tests or lab work?

Would like some help?

Come to the CIS Lab to work with classmates, lab assistants and instructors on Lab assignments.

Rich is in the lab Wednesdays and Fridays from 3:30 - 6:00 PM

Free CIS 90 Tutoring Available

http://www.cabrillo.edu/services/tutorials/

Matt Smithey

All students interested in tutoring in CIS 90, 172, and 81 classes need to come directly to the Tutorials Center to schedule, register and fill out some paperwork. This is just a one-time visit.

The tutoring will take place at the STEM center and they will log in and log out on a computer you have designated (I will figure out exactly what that means).

Matt is available M: 9:00-5:00, T: 9-11 and 2-5, Wed: 9-12 and Th: 9-11 and 3-5.

Housekeeping

Housekeeping

1. Lab 7 due 11:59PM tonight (don't forget to submit it)
2. A **check7** script is available
3. Test #2 is in two weeks (next week is Spring Break)
4. Practice Test #2 available now
5. No lab assigned this week (so you can work on the practice test)
6. Opus will be **unavailable** Sunday evening and all day Monday (3/30-3/31)

Final Exam

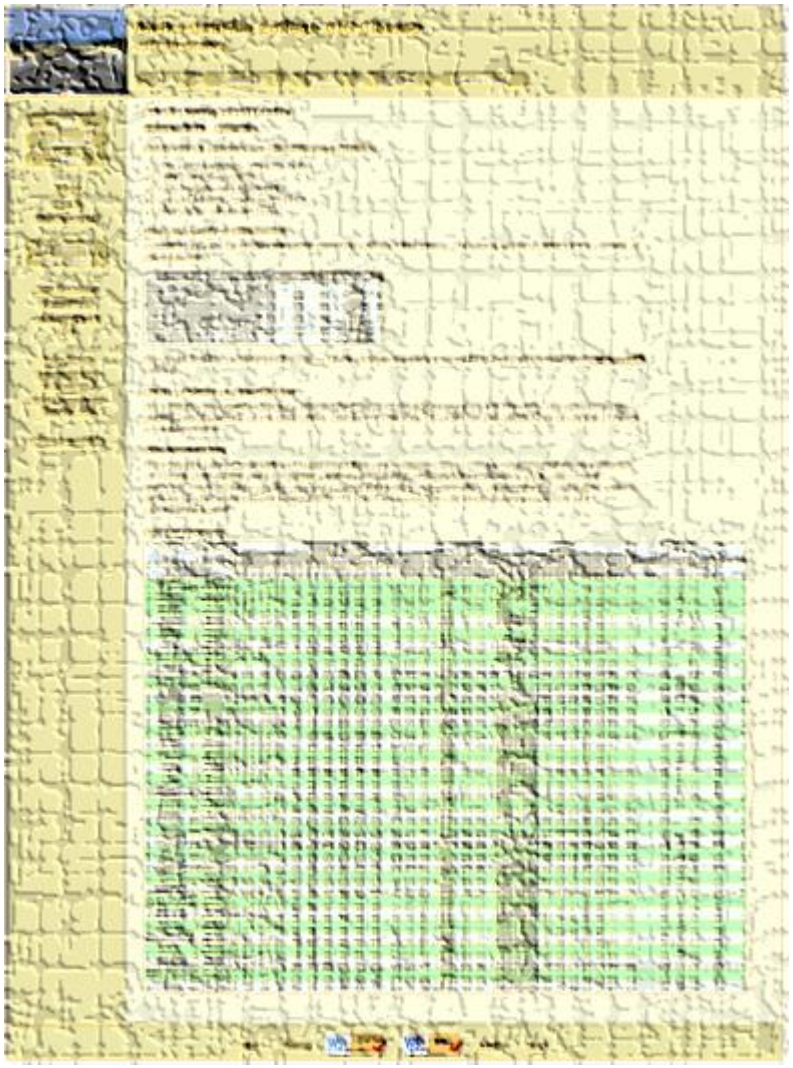
Test #3 (final exam)

- Must be face-to-face or proctored (not online using CCC Confer).
- Room 828 on campus.

	5/21	**Test #3 (the final exam)** **Time** • 7:00AM - 9:50AM in Room 828 **Materials** • Test (download)		5 posts Lab X1 Lab X2

- If you are a long distance student, contact the instructor for options.

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



GRADES

- Check your progress on the Grades page
- If you haven't already, send me a student survey to get your LOR secret code name
- Graded labs & tests are placed in your home directories on Opus
- Answers to labs, tests and quizzes are in the */home/cis90/answers* directory on Opus

Current Point Tally

As of 3/25/2014

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**

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

alatar: 64% (172 of 268 points)
anborn: 78% (210 of 268 points)
aragorn: 88% (237 of 268 points)
arwen: 99% (266 of 268 points)
beregond: 0% (0 of 268 points)
bilbo: 54% (145 of 268 points)
celebrian: 98% (265 of 268 points)
dwalin: 97% (260 of 268 points)
eomer: 91% (245 of 268 points)
faramir: 94% (253 of 268 points)
frodo: 95% (256 of 268 points)
gwaihir: 106% (285 of 268 points)
ioreth: 96% (259 of 268 points)
legolas: 91% (244 of 268 points)
marhari: 65% (176 of 268 points)
orome: 83% (225 of 268 points)
pallando: 0% (0 of 268 points)
pippen: 79% (212 of 268 points)
quickbeam: 99% (267 of 268 points)
rian: 0% (0 of 268 points)
samwise: 85% (230 of 268 points)
shadowfax: 0% (0 of 268 points)
strider: 86% (232 of 268 points)
theoden: 58% (157 of 268 points)
treebeard: 107% (287 of 268 points)
tulkas: 86% (231 of 268 points)
ulmo: 80% (216 of 268 points)

If you are not happy with your current standing contact the instructor ASAP

Jesse's checkgrades python script

http://oslab.cabrillo.edu/forum/viewtopic.php?f=31&t=773&p=2966

```
/home/cis90/simben $ checkgrades smeagol

Remember, your points may be zero simply because the
assignment has not been graded yet.

Quiz 1: You earned 3 points out of a possible 3.
Quiz 2: You earned 3 points out of a possible 3.
Quiz 3: You earned 3 points out of a possible 3.
Quiz 4: You earned 3 points out of a possible 3.


Forum Post 1: You earned 20 points out of a possible 20.

Lab 1: You earned 30 points out of a possible 30.
Lab 2: You earned 30 points out of a possible 30.
Lab 3: You earned 30 points out of a possible 30.
Lab 4: You earned 29 points out of a possible 30.


You've earned 15 points of extra credit.

You currently have a 109% grade in this class. (166 out of
152 possible points.)
```

Use your LOR code name as an argument on the checkgrades command

Jesse is a CIS 90 Alumnus. He wrote this python script when taking the course. It mines data from the website to check how many of the available points have been earned so far.

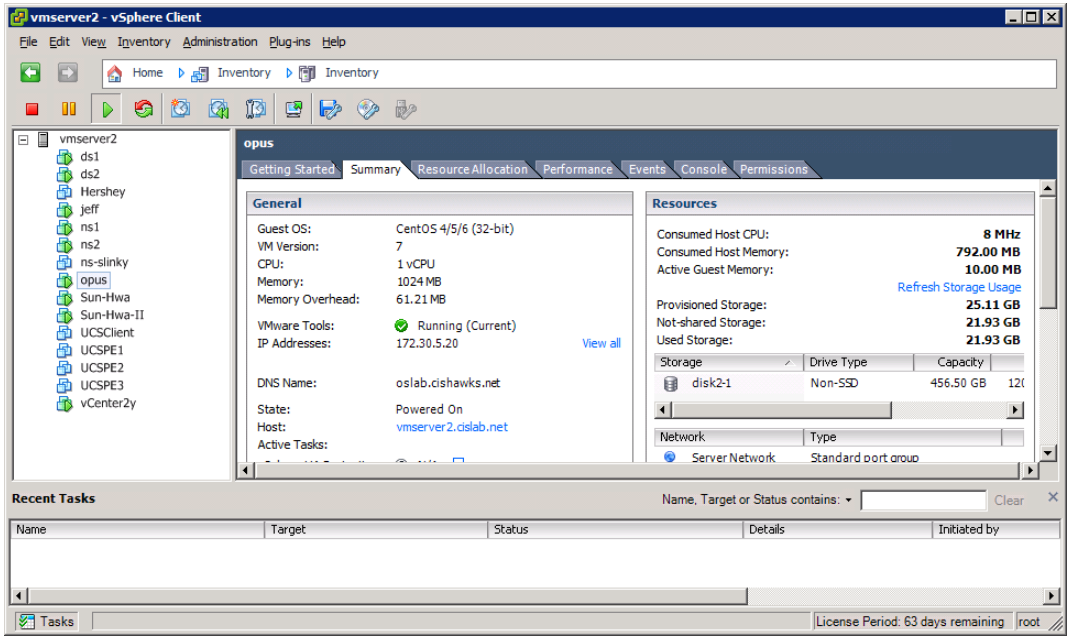
Linux at School

Our Opus server on campus

vmserver2

Opus is a VM running on the vmserver2 server in the CIS Lab

VMs on vmserver2



<u>SSH access to Opus</u>
hostname: oslab.cishawks.net (port 2220)

Your own Linux Systems

USB "Live" Linux Boot USB Drive

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

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 distro of your choice
See: http://iso.linuxquestions.org/

2) 

Get a USB flash drive

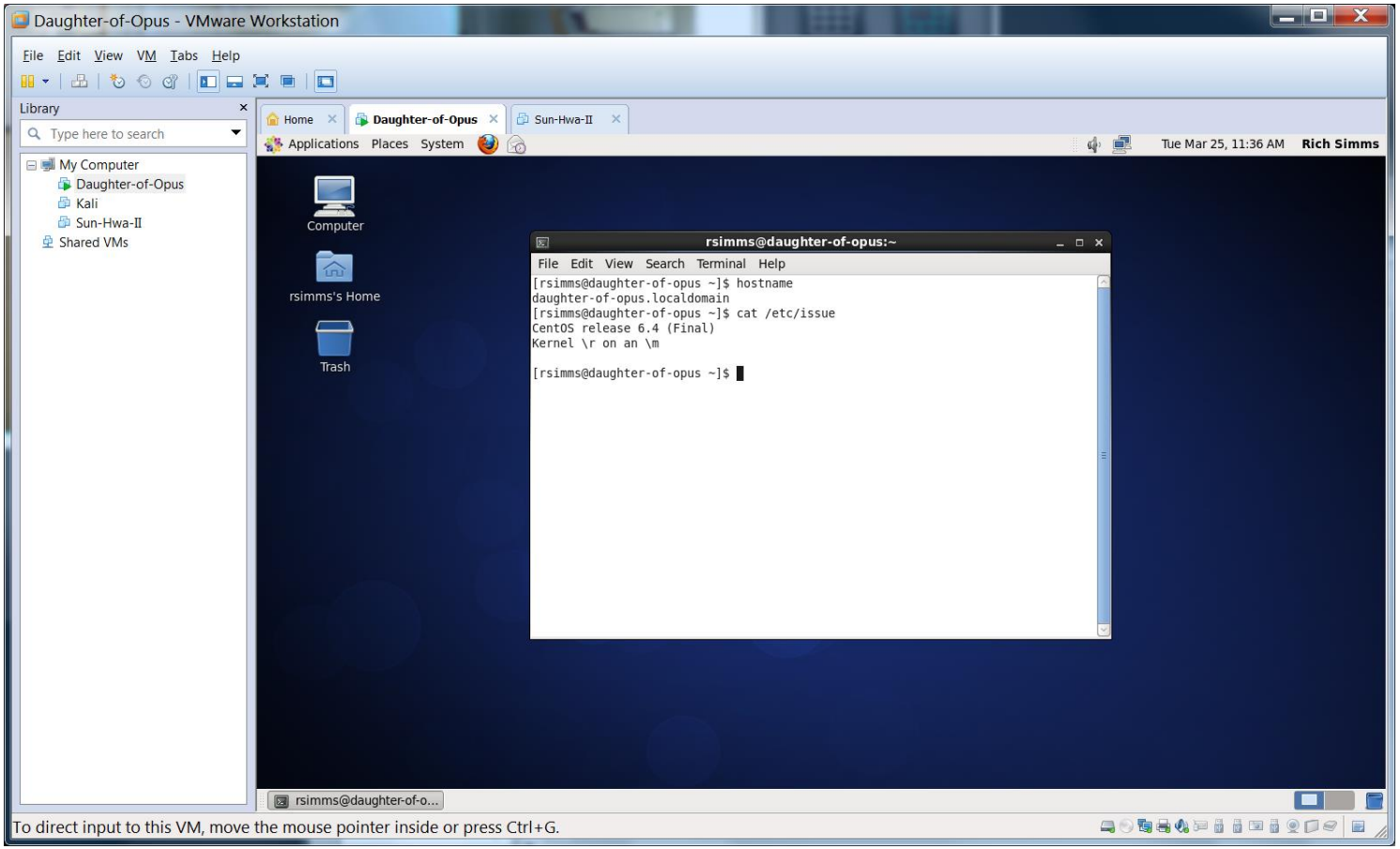
3) Google "boot live linux from usb" for instructions
or see
http://www.howtogeek.com/howto/14912/create-a-persistent-bootable-ubuntu-usb-flash-drive/

4) 

Running native Kali Linux on my Windows laptop
(BIOS configured to boot from USB if present)

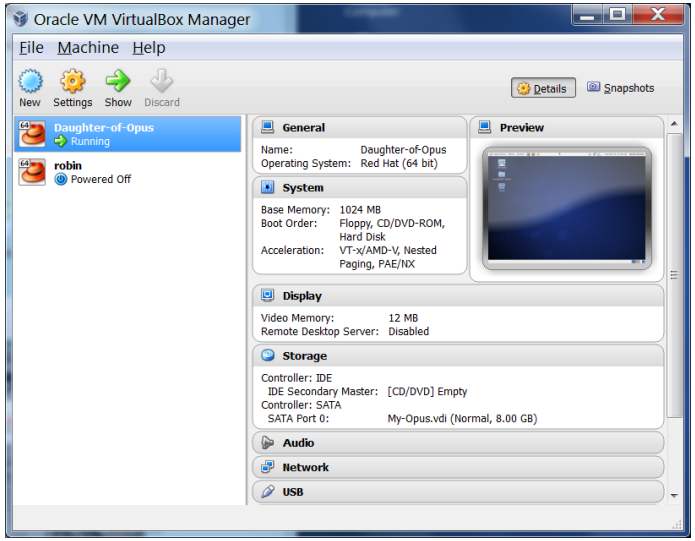
1) Power On with USB stick ==> Kali Linux
2) Power On without USB stick ==> Windows

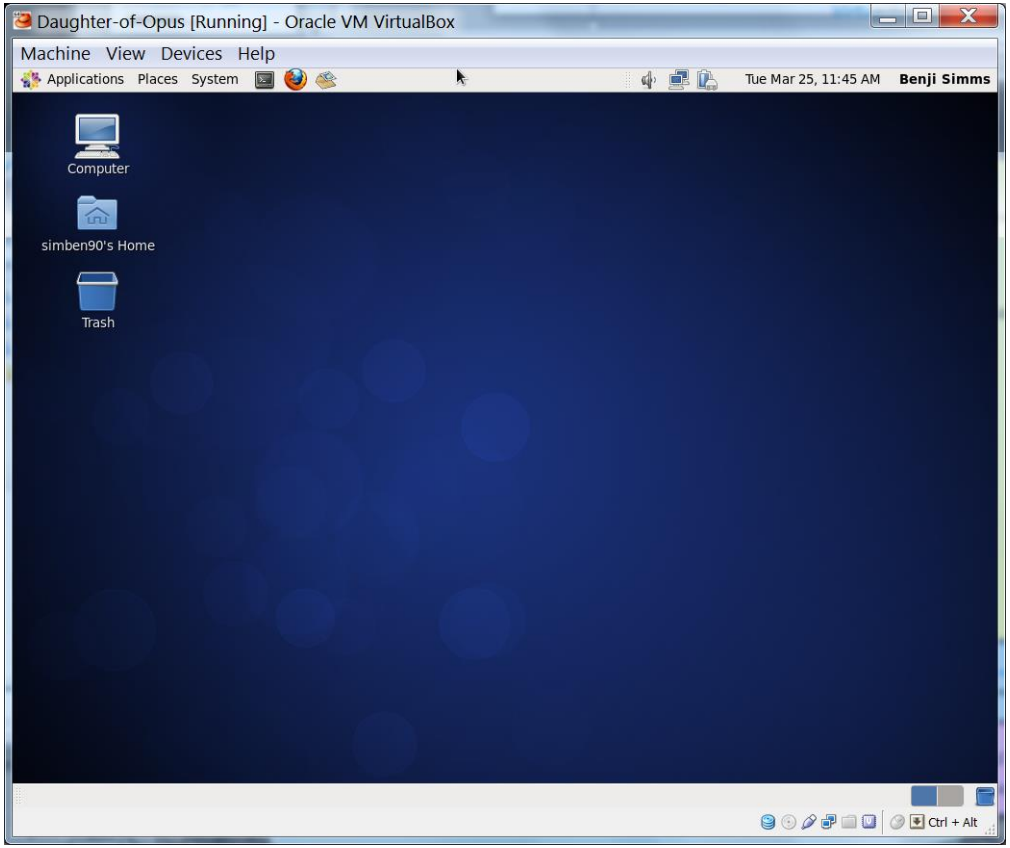
One Daughter-of-Opus
VMware Workstation



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

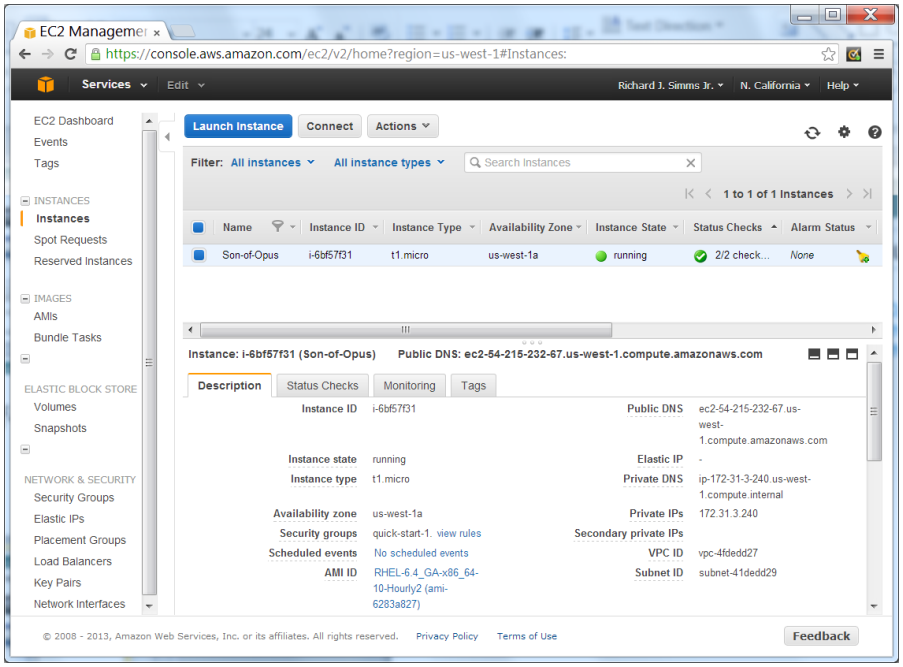
Another Daughter-of-Opus
Oracle VirtualBox





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

Son-of-Opus
Amazon Web Services



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

<u>SSH access to Son-of-Opus</u>
hostname: son-of-opus.simms-teach.com (port 2220)

Baby-Opus
Debian 7 Linux Server

Raspberry Pi

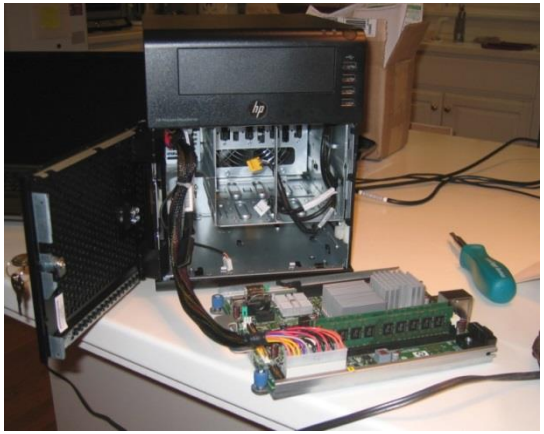
Baby-Opus is a VM running on my Raspberry Pi

SSH access to Baby-Opus
hostname: <*ip-address*> (port 22)

NoPar#show ip dhcp binding
MAC b8:27:eb:b7:b3:99
Reservation for 172.30.1.31

My Home VLab

HP Microserver

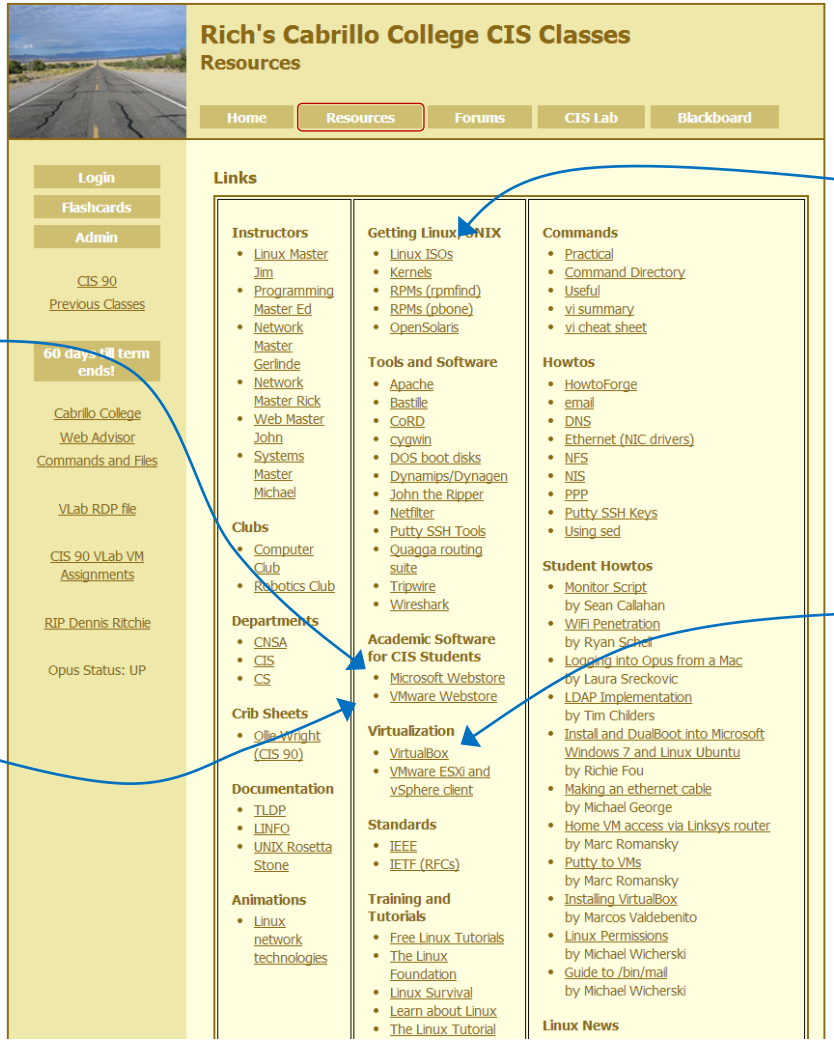


VMware ESXi for virtualization

Inexpensive "bare bones" servers are available that come without hard drives or an operating system

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*.

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*

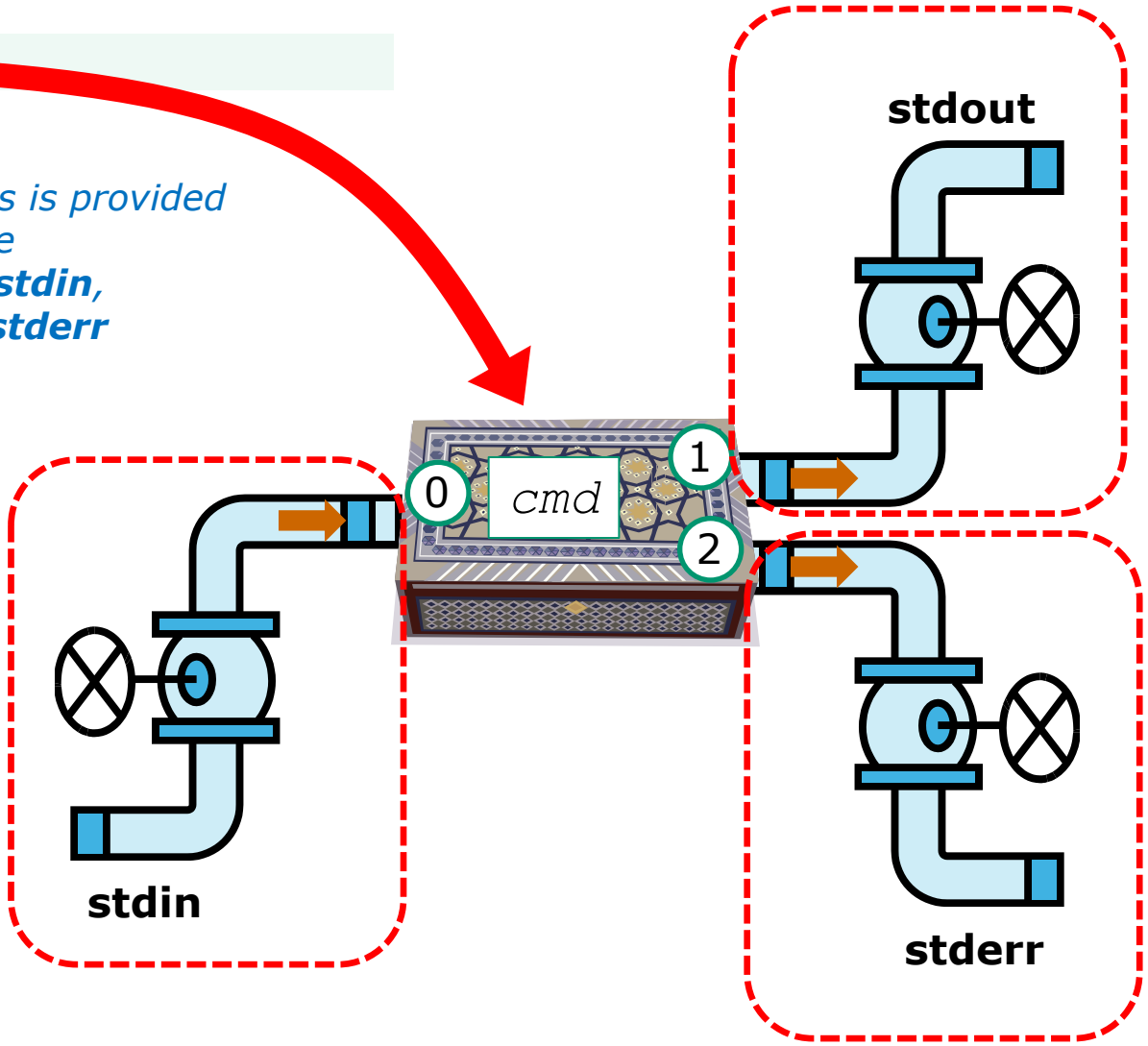
Examples

```
/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**

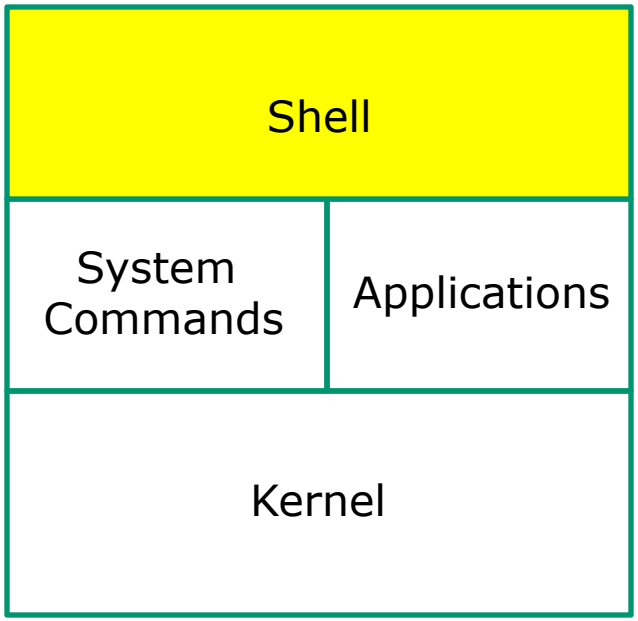
```
$ cmd
```

*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

Example

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

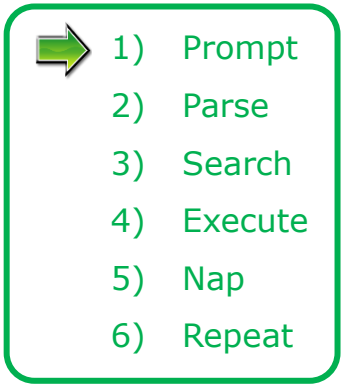
The shell begins by echoing a **prompt** string to your terminal device:

- 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.

```
/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.

Example



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

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

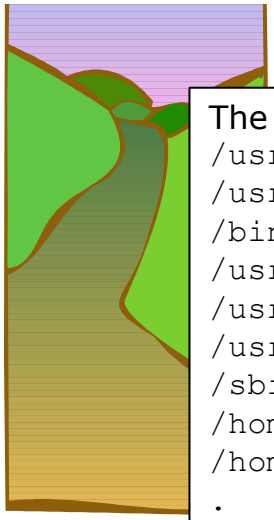
1) Prompt
2) Parse
3) Search
4) Execute
5) Nap
6) Repeat

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.



The ***Path*** (**echo $PATH** to show)

```
/usr/lib/qt-3.3/bin:
/usr/local/bin:
/bin:
/usr/bin:
/usr/local/sbin:
/usr/sbin:
/sbin:
/home/cis90/simben/../bin:
/home/cis90/simben/bin:
.
```

`sort`

The shell locates the sort command in the /bin directory which is the third directory of a CIS 90 student's path.

Example

argument

option

redirection

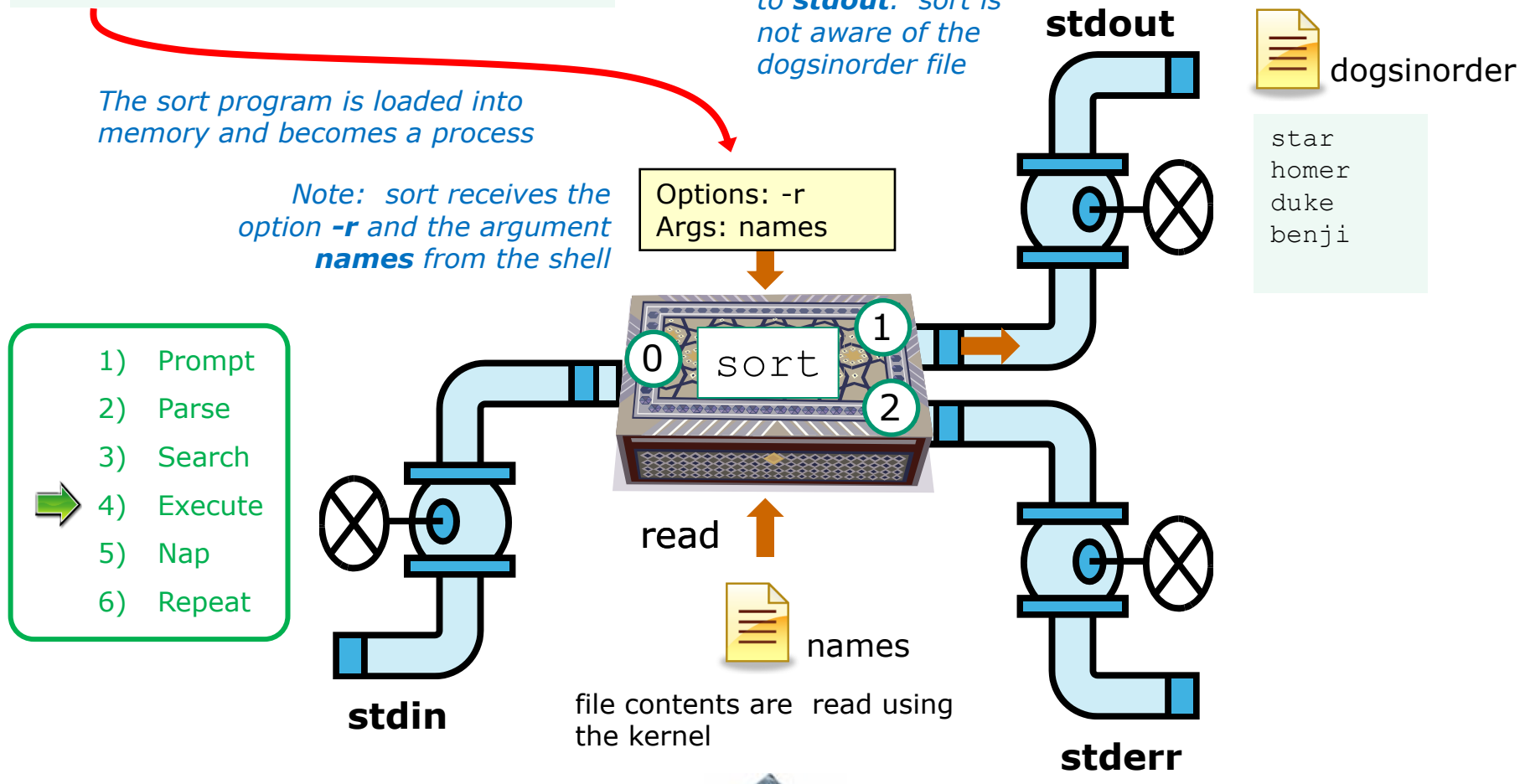
```
$ sort -r names > dogsinorder
```

*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*



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

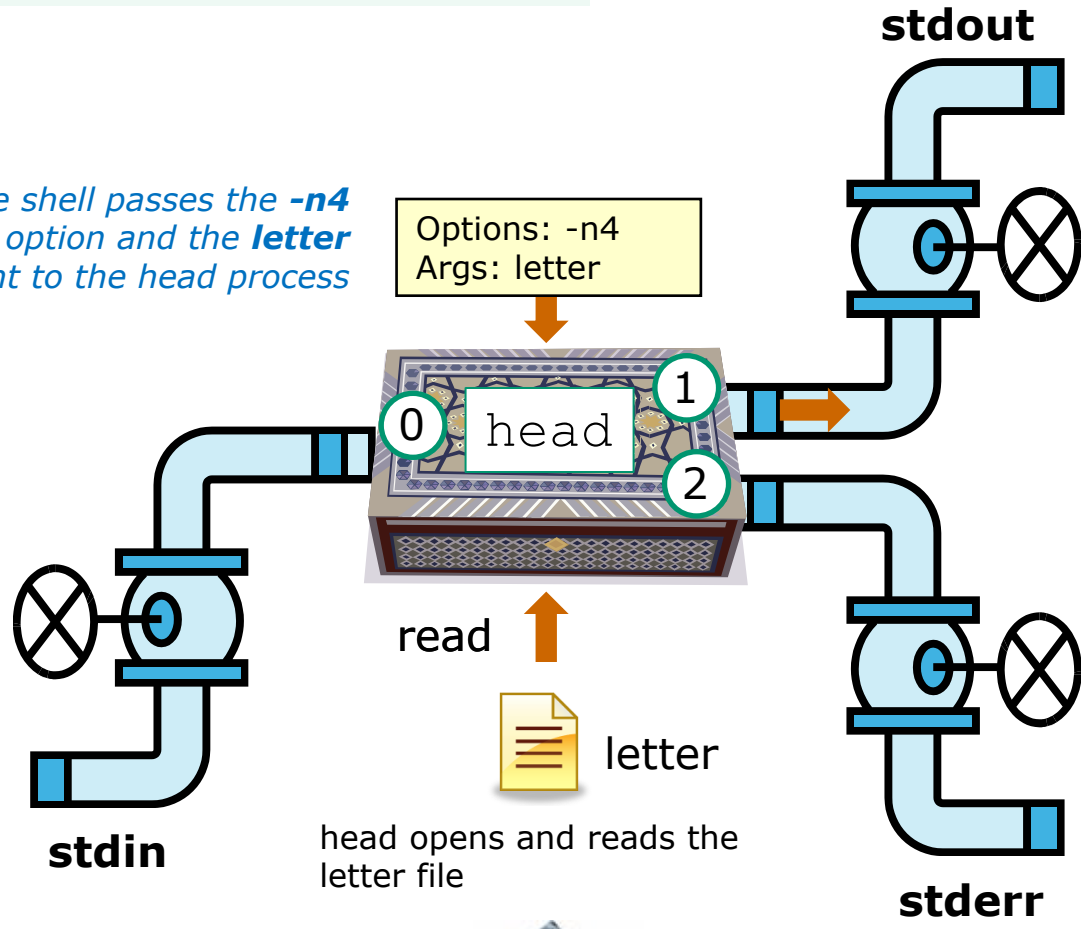
```
/home/cis90/simben $ head -n4 letter
Hello Mother!  Hello Father!

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

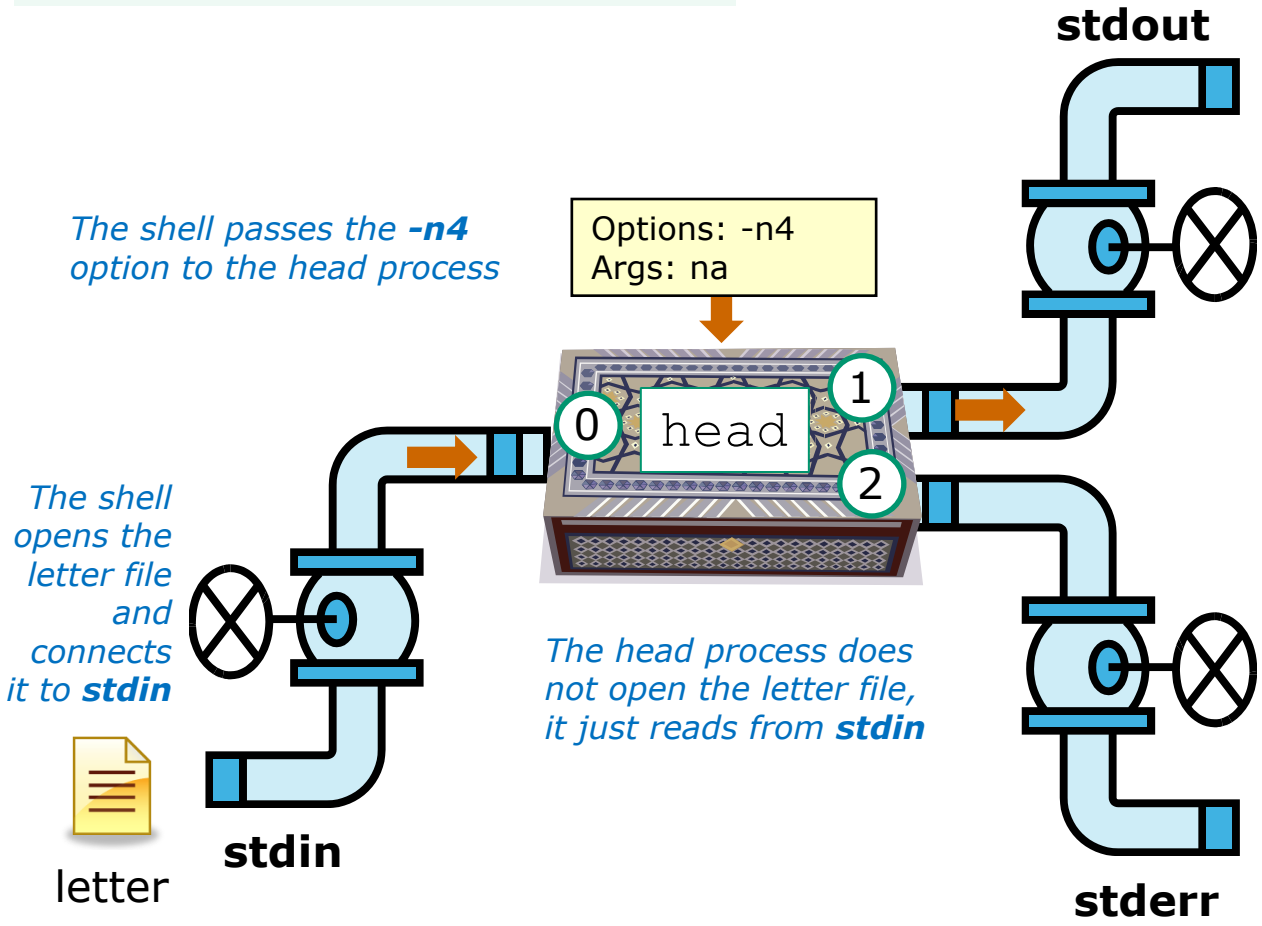
head opens and reads the letter file

head -n4 < letter

option

redirection

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



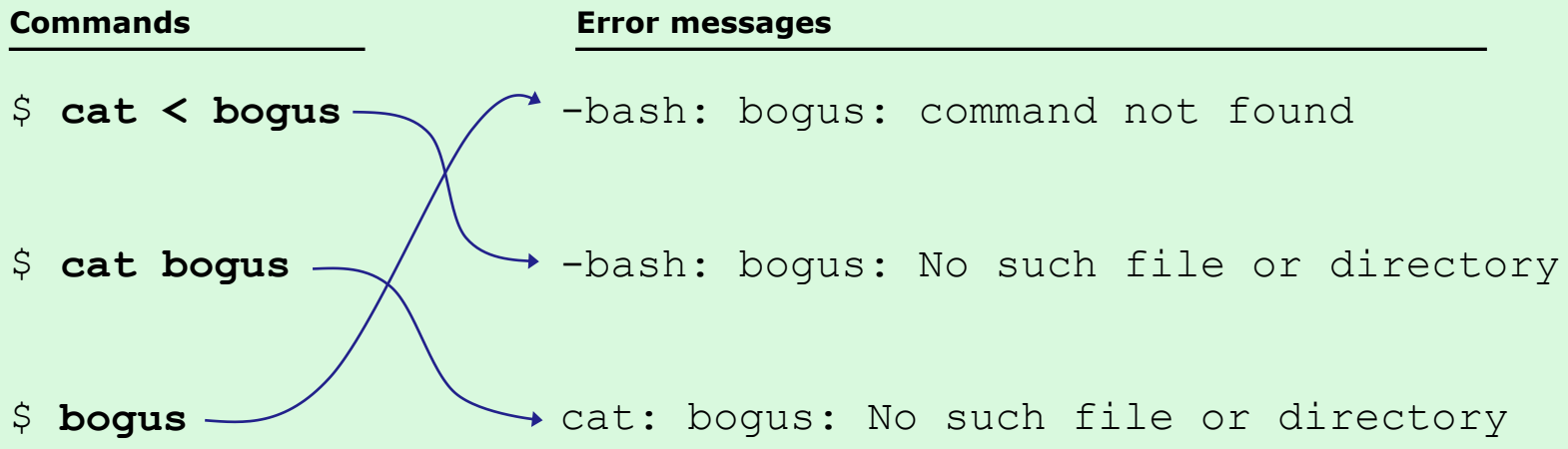
Test your understanding of how the shell and command work as a team

Given: There is no file named *bogus*, associate each command on the left with an error message on the right

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`

Test your knowledge

Given: There is no file named bogus, associate each command on the left with an error message on the right



2>&1

FYI

(more on this in CIS 98)

It's descriptor clobbering time!

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

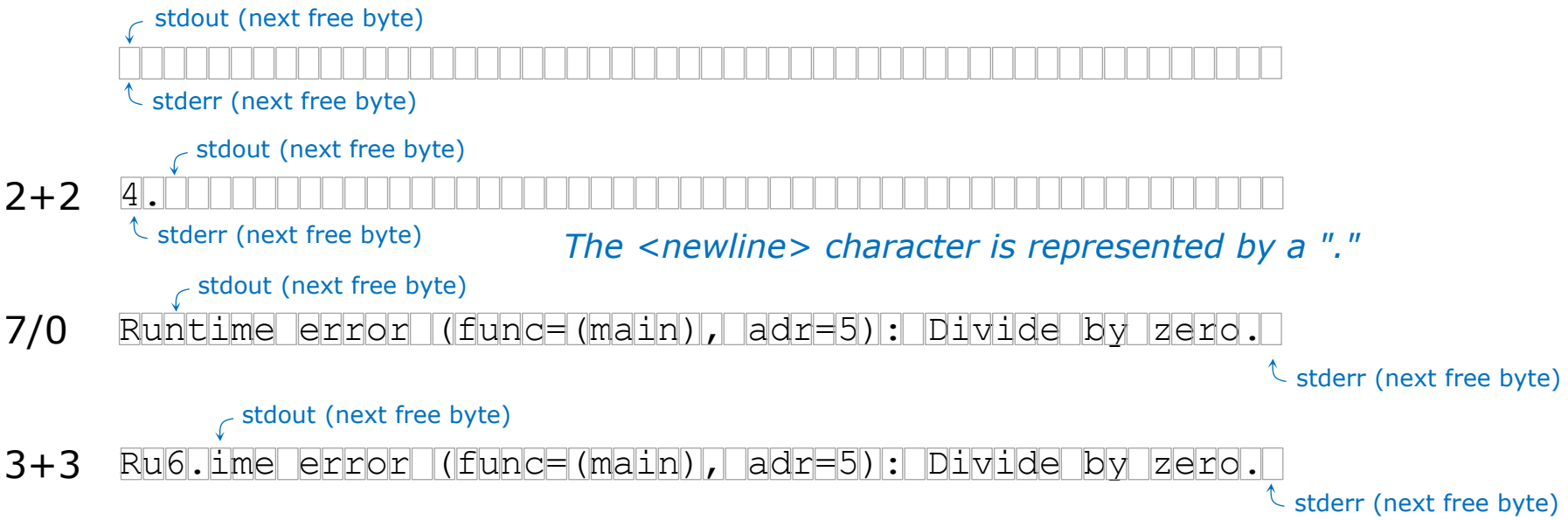
/home/cis90/simben $ **cat calculations**
Ru6
ime error (func=(main), adr=5): Divide by zero

*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!

```
/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)
C program example

C Program I/O example

```
[rsimms@opus misc]$ 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);
}
```

This program is available in the depot directory

C Program I/O example

```
[rsimms@opus misc]$ 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***

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

C Program I/O example

```
[rsimms@opus misc]$ 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);
}
```

*Read users name from **stdin***

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

C Program I/O example

```
[rsimms@opus misc]$ 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 greeting to **stdout***

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

C Program I/O example

```
[rsimms@opus misc]$ 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 users name to **stdout***

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

C Program I/O example

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

```
[rsimms@opus misc]$ 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

```
[rsimms@opus misc]$ ./simple
What is your name stranger? Rich
Well I'm very pleased to meet you, Rich
```

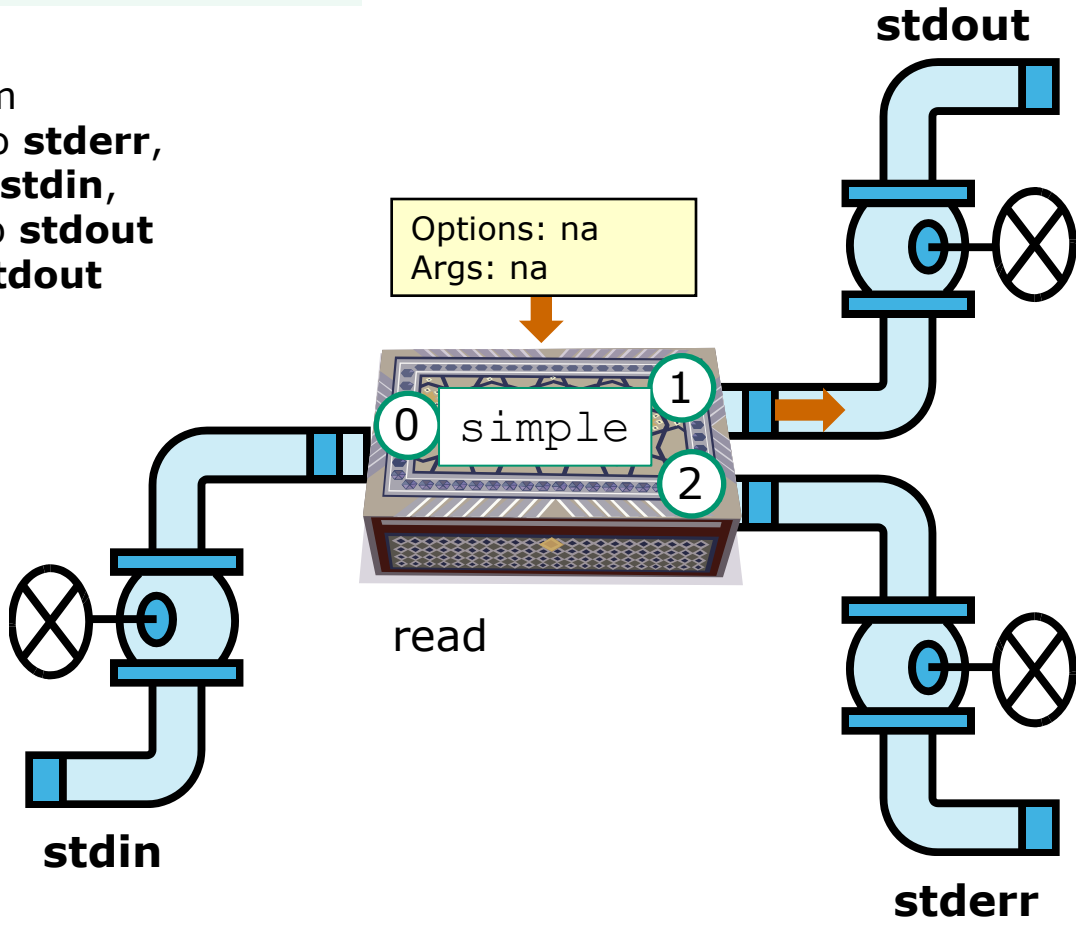
Running the simple program.

*Note I need to preface **simple** with a "**./**" to run it as this directory is not on my path. This is not necessary for CIS 90 students as they already have the . directory in their path.*

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.

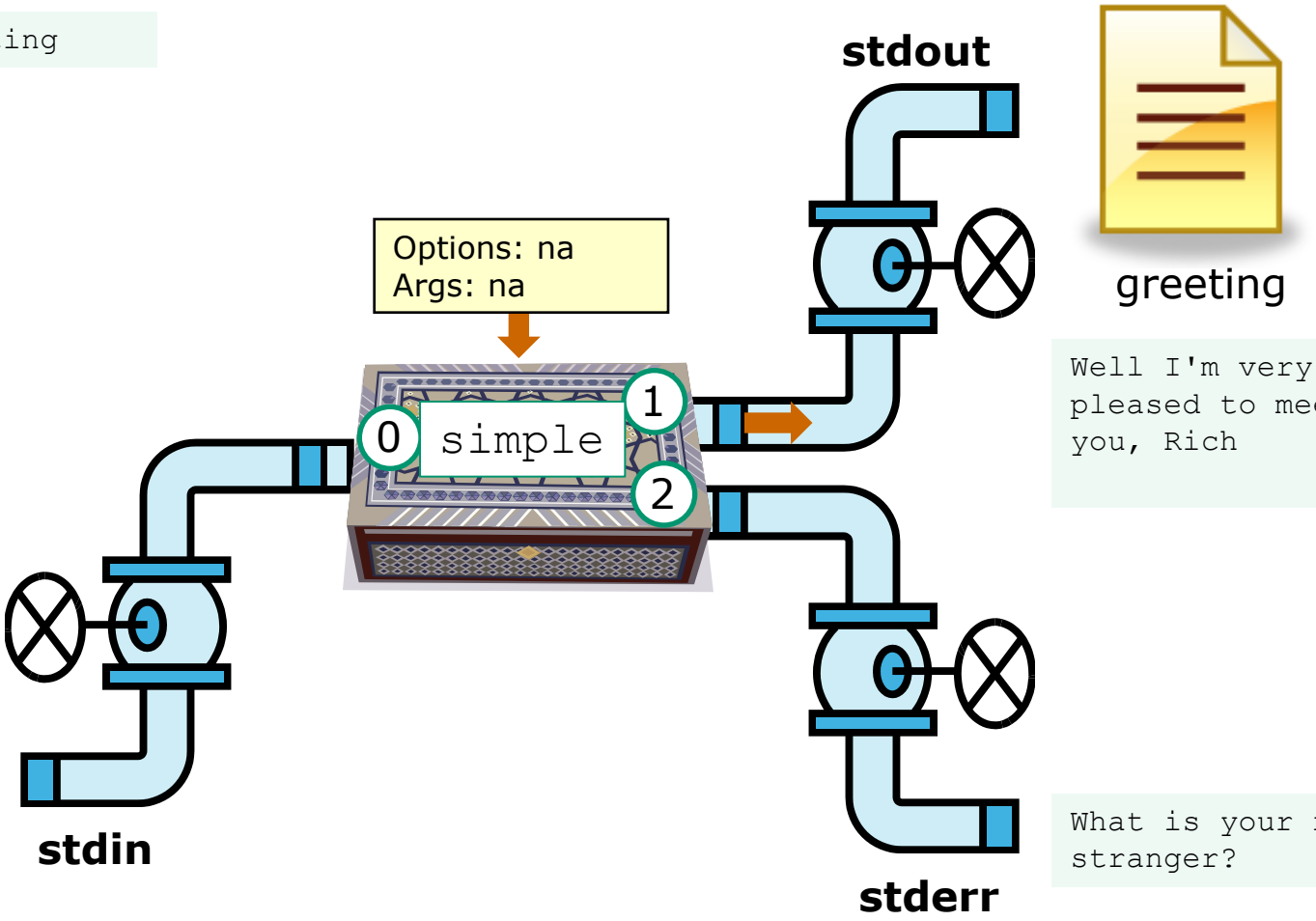
```
[rsimms@opus misc]$ ./simple  >  myfile
What is your name stranger? Rich

[rsimms@opus misc]$ 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 simple.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

More on umask

(review)

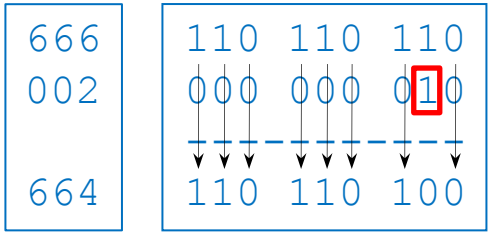
Review - applying umask bits

Current umask setting

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

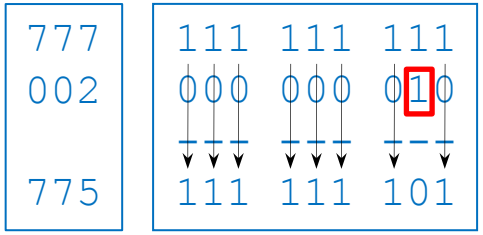
this mask indicates which permissions should NOT be set on the new file or directory

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

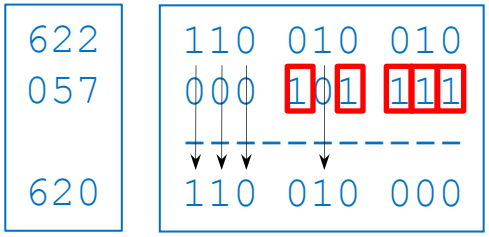



```
/home/cis90/simben/lesson9 $ mkdir newdir
/home/cis90/simben/lesson9 $ ls -ld newdir
drwxrwxr-x 2 simben cis90 4096 Oct 27 07:23 newdir
```

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
```



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.*

More Pipeline Practice

Pipelines

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

grep Homer /etc/passwd
milhom90:x:1202:190:Homer Miller:/home/cis90/milhom:/bin/bash

last
last | grep milhom90
last | grep milhom90| grep "Mon"
last | grep milhom90| grep "Mon" | tee mylog
cat mylog
last | grep milhom90| 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

More Pipeline Practice

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.

More Pipeline Practice

Pipelines

Task
Print a sorted list of the first names for CIS 172 students

```
cat /etc/passwd
cat /etc/passwd | grep cis172
cat /etc/passwd | grep cis172 | cut -f 5 -d ":"
cat /etc/passwd | grep cis172 | cut -f 5 -d ":" | cut -f1 -d" "
cat /etc/passwd | grep cis172 | 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 ?????
cat /etc/?????? | grep ????? | cut -f ? -d "?"
cat /etc/?????? | grep ????? | cut -f ? -d "?" | cut -f? -d"?" | ????

Put your list in the chat window.

More on pipelines

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

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

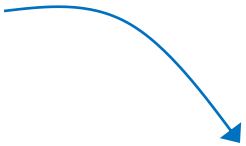
```
/home/cis90/simben $ head -n2 poems/Anon/nursery
Jack and Jill went up the hill
to fetch a pail of water.
/home/cis90/simben $ head -n2 poems/Anon/nursery | wc -l
2
/home/cis90/simben $
```

*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

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

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

Eggs

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 CCC Confer next to your name when you have collected 3 eggs, electronically "clap" if you collect all 17.

Review

Jim's Summary Pages

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

Lesson 6 - Managing Files
http://cabrillo.edu/~jgriffin/CIS90/files/lecture5.html

Lesson 7 - File Permissions
http://cabrillo.edu/~jgriffin/CIS90/files/lecture6.html

Lesson 8 - Input/Output Processing
http://cabrillo.edu/~jgriffin/CIS90/files/lecture7.html

Make Teams

Breakout Rooms

Room 1

Room 2

Room 3

Room 4

Room 5

Once you are in your rooms:

1) Write your team's distro name at the top of your room's white board
2) Everyone write their first names under the distro's team name
3) If you want to be fancy add your distro logo to the top of your room's white board!

Make Teams:
CCC Confer: Tools > Breakout Rooms > Create Breakout Rooms … (make 6 rooms)

Flashcard Practice

Flashcards

Room 1	Room 2	Room 3	Room 4	Room 5
Points:	**Points:**	**Points:**	**Points:**	**Points:**

Flashcards
L6=20
L7=15
L8=16

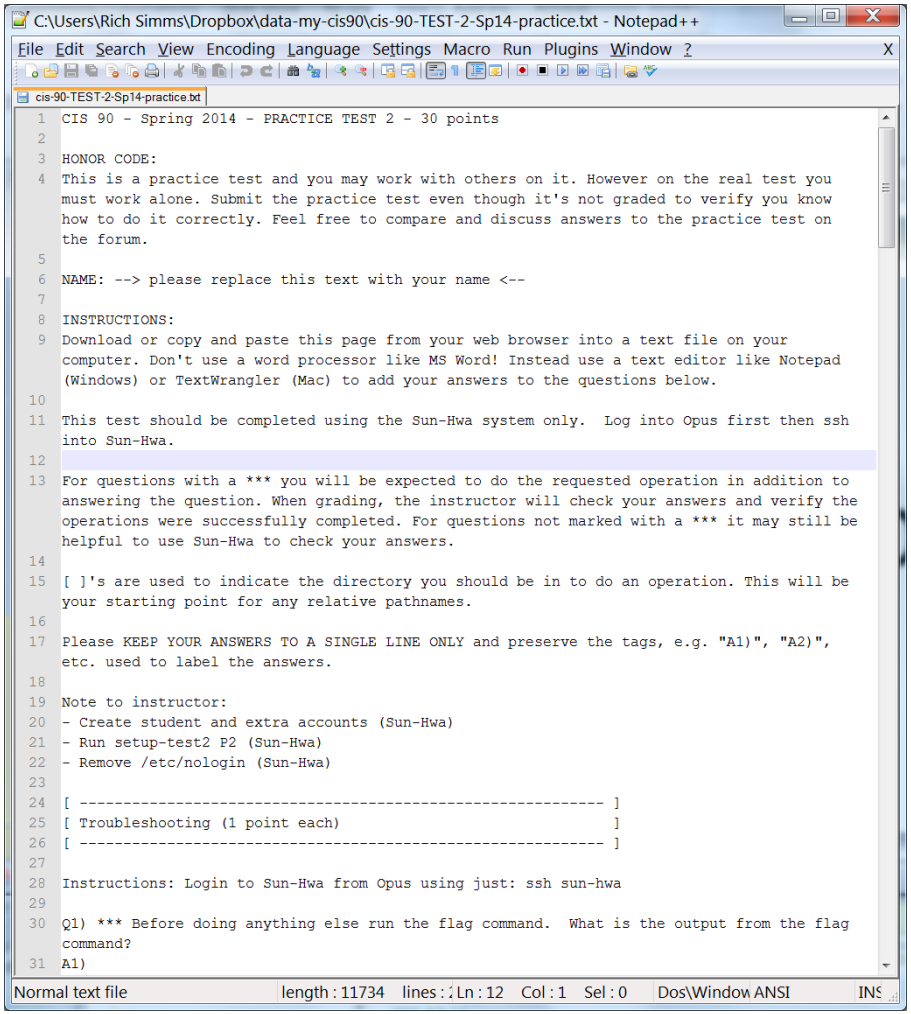
Rules

- 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

Practice Test

Practice Test



Practice test available

- Available on the website
- Work alone or together
- 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.

```
 8  INSTRUCTIONS:
 9  Download or copy and paste this page from your web browser into a text file on your
    computer. Don't use a word processor like MS Word! Instead use a text editor like Notepad
    (Windows) or TextWrangler (Mac) to add your answers to the questions below.
10
11  This test should be completed using the Sun-Hwa system only.  Log into Opus first then ssh
    into Sun-Hwa.
12
13  For questions with a *** you will be expected to do the requested operation in addition to
    answering the question. When grading, the instructor will check your answers and verify the
    operations were successfully completed. For questions not marked with a *** it may still be
    helpful to use Sun-Hwa to check your answers.
14
15  [ ]'s are used to indicate the directory you should be in to do an operation. This will be
    your starting point for any relative pathnames.
16
17  Please KEEP YOUR ANSWERS TO A SINGLE LINE ONLY and preserve the tags, e.g. "A1)", "A2)",
    etc. used to label the answers.
18
```

```
217 [ ---------------------------------------------------------- ]
218 [ Submit this test                                           ]
219 [ ---------------------------------------------------------- ]
220
221 When finished, leave your home directory intact. Copy and paste this completed test into a
    text-only email with NO attachments to:
222
223   rsimms@oslab.cabrillo.edu
224   <your-opus-username>@oslab.cabrillo.edu
225
226 CONFIRM on Opus that your email was successfully sent and that your work is READABLE using
    the mail command.
227
228
```

Wrap up

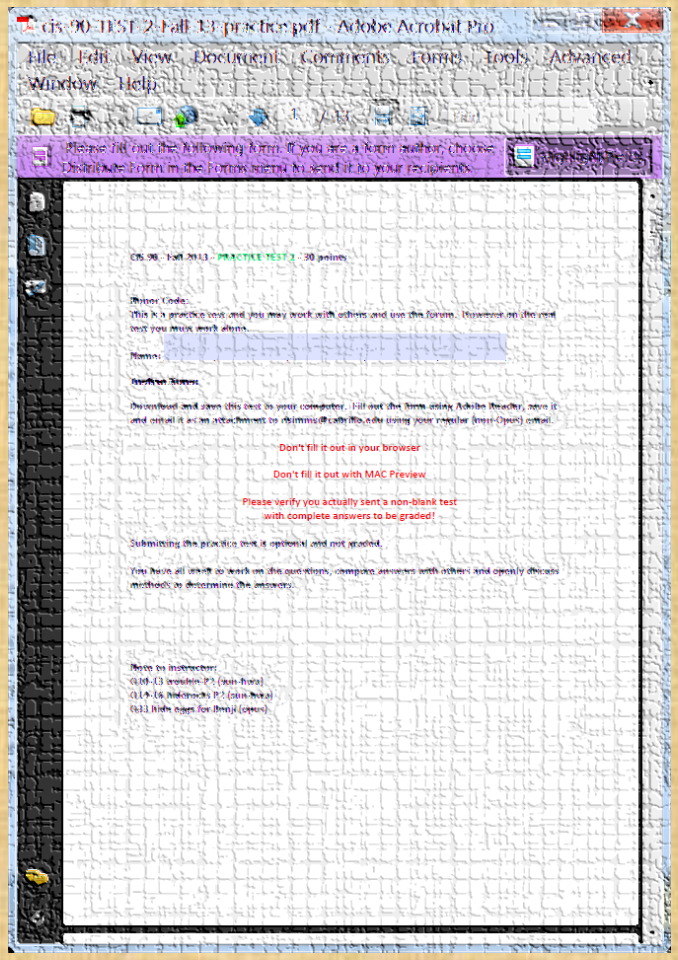
Next Class

No Quiz

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

- Recommended preparation:
  - **Work the practice test!**
  - **Work the practice test!**
  - **Work the practice test!**
  - **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

Optional Workshop Today



Work the practice test till the end of class today

- Collaborate!
- Ask questions!
- You may leave class once you know how to approach and hopefully answer each question

Backup