



## Rich's lesson module checklist

- Slides
- Converted WB
  
- Flash cards
- Page numbers
- 1<sup>st</sup> minute quiz
- Web Calendar summary
- Web book pages
- Commands
  
- Practice test tested
- BlackBoard test replicated to both sections
- Primary and secondary practice test servers up and logins enabled
- Q29 email script tested and scheduled to send at end of Lesson 5
  
- Schedule lock of turnin directory and submit
  - at 12:00 am Thursday  
chmod 700 /home/cis90/bin/submit  
chmod 700 /home/turnin/cis90  
ctrl-d
  - at 9:00 am thursday  
chmod 750 /home/cis90/bin/submit  
chmod 755 /home/turnin/cis90  
ctrl-d
- Lesson 5 troubleshooting server ready and logins enabled
- Flash cards and timer script ready
  
- 9V backup battery for microphone
- Backup slides, CCC info, handouts on flash drive



### **Student Learner Outcomes**

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

## Introductions and Credits



Jim Griffin

- Created this Linux course
- Created Opus and the CIS VLab
- Jim's site: <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 for laying out screen when attending class

- Browse to the CIS 90 website Calendar page
  1. <http://simms-teach.com>
  2. Click CIS 90 link on left panel
  3. Click Calendar link near top of content area
  4. Locate today's lesson on the Calendar
  
- Download the presentation slides for today's lesson for easier viewing
  
- Click Enter virtual classroom to join CCC Confer session
  
- Connect to Opus using Putty or ssh command



## Student checklist for laying out screen when attending class

Google

CCC Confer

Downloaded PDF of Lesson Slides

The screenshot shows a virtual classroom interface with several overlapping windows:

- Blackboard Course Page:** Displays 'Rich's Cabrillo College CIS 90 Calendar' with a sidebar containing navigation links like 'Login', 'Flashcards', 'Admin', and 'CIS 90 (Spring) Course Home'.
- CCC Confer Virtual Classroom:** The main window showing a video feed of 'Rich Simms' and a 'PARTICIPANTS' list with 'Rich Simms' and 'Benji Simms'. A 'CHAT' window shows a conversation about textbooks.
- Google Maps:** A map window titled 'Cabrillo College' showing the campus location.
- PDF Viewer:** A window titled 'cis90lesson01.pdf - Adobe Acrobat Pro' displaying 'The CIS 90 System Playground' slide.
- Terminal Window:** A terminal window showing login prompts for 'Opus' and 'Rich Simms' with timestamps and IP addresses.

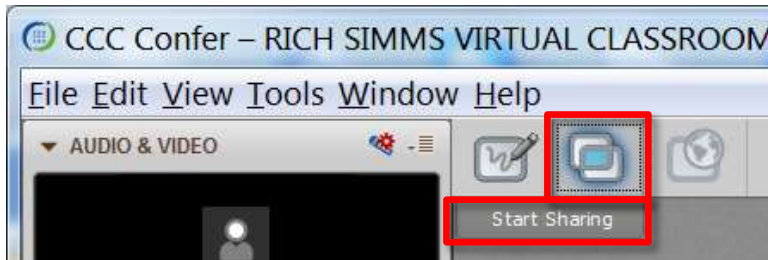
CIS 90 website Calendar page

One or more login sessions to Opus

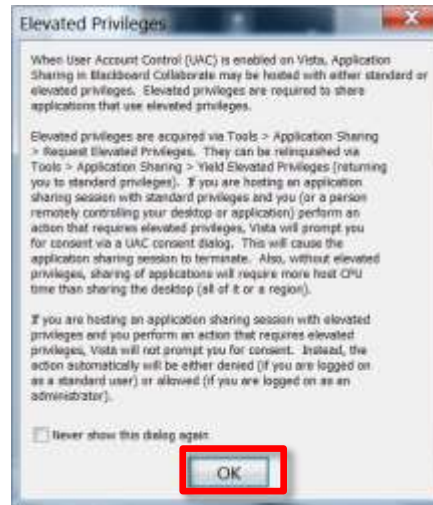


## Student checklist for sharing desktop with classmates

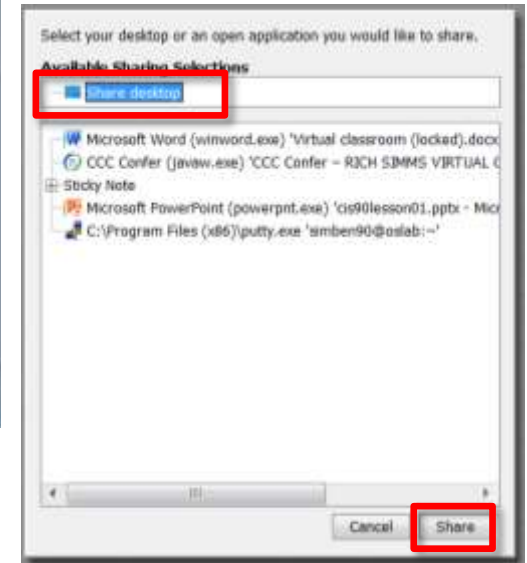
1) Instructor gives you sharing privileges



2) Click overlapping rectangles icon. If white "Start Sharing" text is present then click it as well.



3) Click OK button.



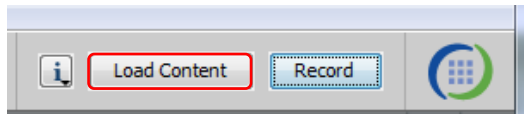
4) Select "Share desktop" and click Share button.



## Rich's CCC Confer checklist - setup

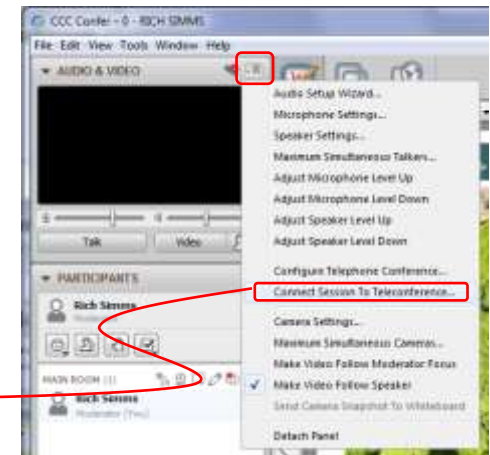


[ ] Preload White Board

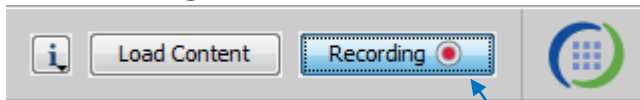


[ ] Connect session to Teleconference

*Session now connected to teleconference*



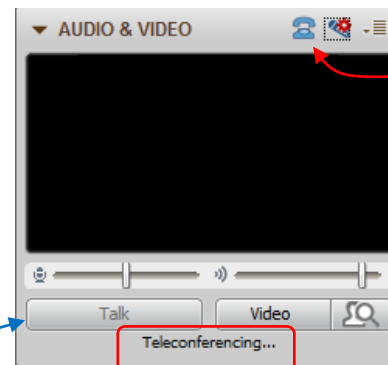
[ ] Is recording on?



*Red dot means recording*

[ ] Use teleconferencing, not mic

*Should be greyed out*



*Should show as this live "off hook" telephone handset icon and the Teleconferencing ... message displayed*



## Rich's CCC Confer checklist - screen layout and share



Rich's CCC Confer checklist - screen layout and share

Annotations in the screenshot:

- foxit for slides
- chrome
- putty
- vSphere Client

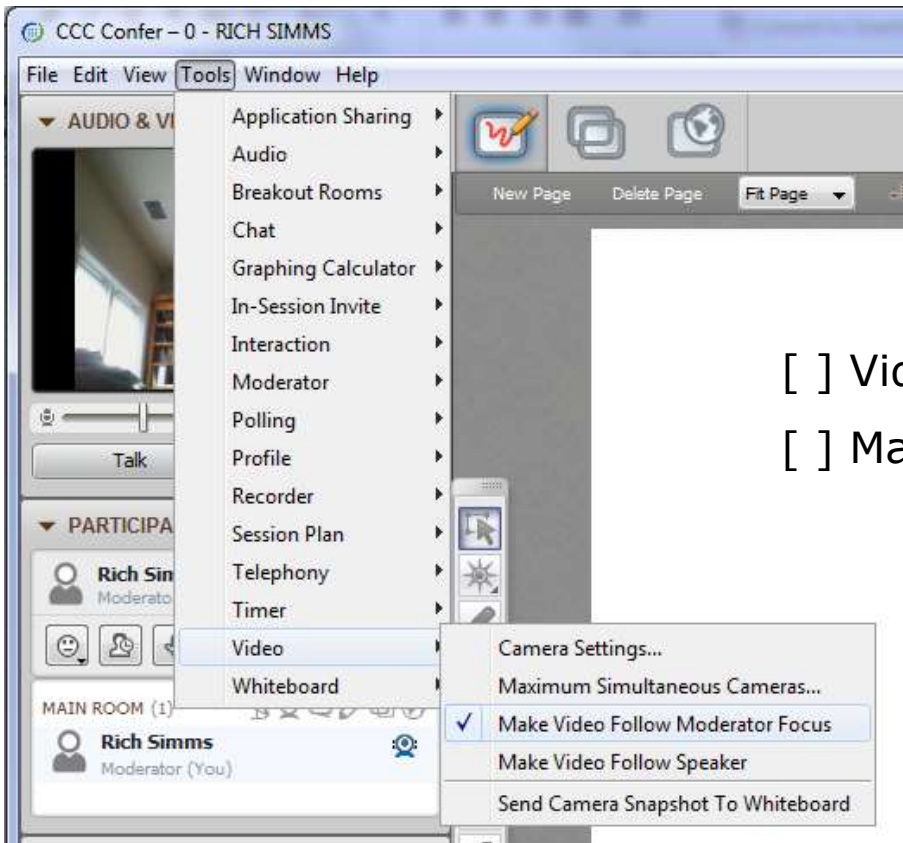
[ ] layout and share apps







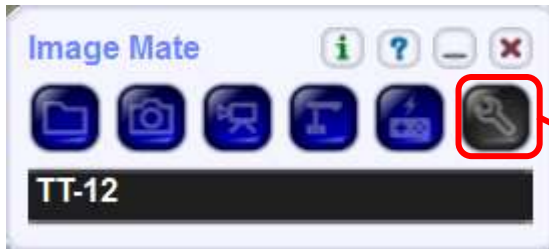
## Rich's CCC Confer checklist - webcam setup



- [ ] Video (webcam)
- [ ] Make Video Follow Moderator Focus



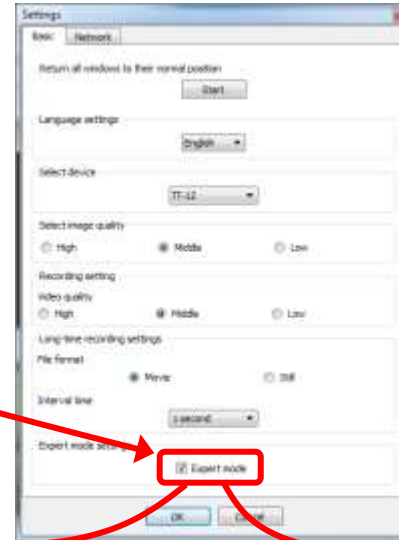
### Rich's CCC Confer checklist - Elmo



Elmo rotated down to view side table



Run and share the Image Mate program just as you would any other app with CCC Confer



The "rotate image" button is necessary if you use both the side table and the white board.

Quite interesting that they consider you to be an "expert" in order to use this button!

Elmo rotated up to view white board





**Rich's CCC Confer checklist - universal fix**

Universal Fix for CCC Confer:

- 1) Shrink (500 MB) and delete Java cache
- 2) Uninstall and reinstall latest Java runtime
- 3) <http://www.cccconfer.org/support/technicalSupport.aspx>

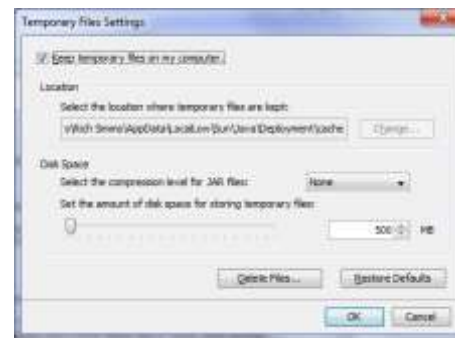
Control Panel (small icons)



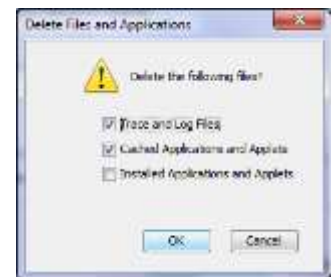
General Tab > Settings...



500MB cache size



Delete these



Google Java download





# Start

# Sound Check

*Students that dial-in should mute their line using \*6 to prevent unintended noises distracting the web conference.*

*Instructor can use \*96 to mute all student lines.*



Instructor: **Rich Simms**

Dial-in: **888-886-3951**

Passcode: **136690**



Chris



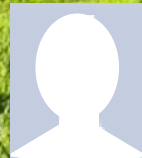
Jeremy



Jennifer



Cameron



Joseph



Lisa



May



Sundance



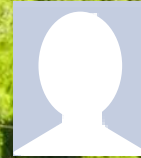
Charlie



Sean



Brenda



Anthony



Will H.



Josh



Michael



Danny



Vic



William D.



Taylor



Thomas



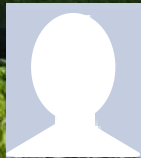
Stewart



Miguel



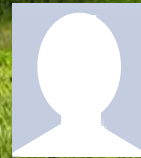
Akasha



Jairo



Tony



Joaquin

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

## First Minute Quiz

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

Use CCC Confer White Board

**email answers to: [risimms@cabrillo.edu](mailto:risimms@cabrillo.edu)**

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

# Review

## Objectives

- Review Lessons 1-4
- Practice skills
- Learn about filename expansion characters

## Agenda

- Quiz
- Questions
- Six steps of shell (review)
- Trouble on the island
- Housekeeping
- Everything is a file
- Filename expansion (globbing)
- Filename expansion practice
- Command review
- Command line syntax & parsing (review)
- Command line syntax & parsing practice
- Metacharacters (review)
- Environment variables (review)
- Inputs & outputs (continuing)
- Architecture (review)
- File system (review)
- CCC Confer
- Flashcards
- Test tips
- Assignment
- Wrap up





# Questions

# Questions

Lesson material?

Labs?

How this course works?

*Are you enlightened yet?*



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.*

**September 27, 2015**





# Six Steps of the shell

(review)

## Which shell are you using?

```
/home/cis90/simben/Poems/Yeats $ ls /bin/*sh  
/bin/bash /bin/csh /bin/dash /bin/ksh /bin/rbash /bin/sh /bin/tcsh
```

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

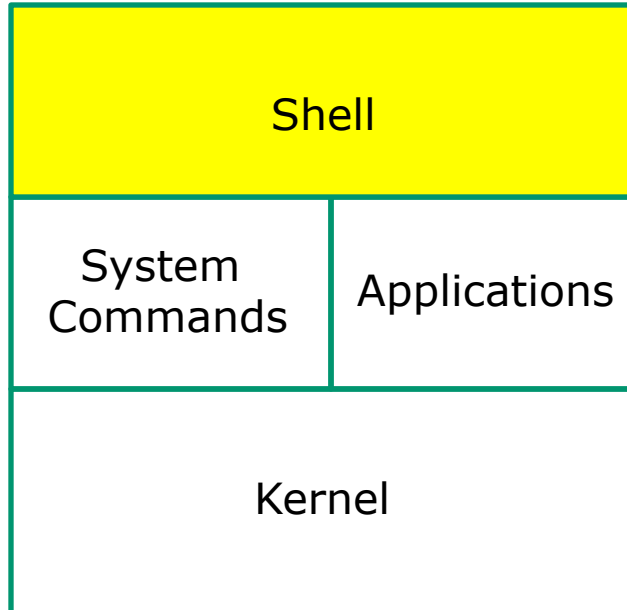
```
/home/cis90/simben/Poems/Yeats $ ps  
PID TTY      TIME CMD  
4635 pts/0    00:00:00 bash  
4785 pts/0    00:00:00 ps
```

```
/home/cis90/simben/Poems/Yeats $ echo $SHELL  
/bin/bash
```

*There are many shells on Opus. They can be found in the /bin directory. Your account entry in /etc/passwd determines which shell you will use.*



# Life of the Shell



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



## An example or how the shell and a command work together as a team to get things done

```
/home/cis90/simben/Poems/Yeats $ file *
mooncat:      ASCII English text
old:          ASCII English text
whitebirds:  ASCII English text
```

Step	Bash shell /bin/bash	File command /usr/bin/file
Prompt	✓	
Parse	✓	
Search	✓	
Execute	✓	✓
Nap		✓
Repeat	✓	


*This table indicates for each step whether bash and/or the command is running*

# 1) Prompt - the shell prompts user for a command

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

*Every time you hit the Enter key the shell will prompt you for another command*

```
/home/cis90/simben/Poems/Yeats $  
/home/cis90/simben/Poems/Yeats $  
/home/cis90/simben/Poems/Yeats $  
/home/cis90/simben/Poems/Yeats $  
/home/cis90/simben/Poems/Yeats $ file *
```





## 1) Prompt - the shell prompts user for a command

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

```
/home/cis90/simben/Poems/Yeats $ echo $PS1
$PWD $
  _ _
```

*The shell uses the value of the PS1 variable to make the prompt.*

```
/home/cis90/simben/Poems/Yeats $ echo $PWD $
/home/cis90/simben/Poems/Yeats $
```

*You can manually do the same thing by echoing the value of the PS1 variable.*

FYI, your PS1 variable on Opus gets set when you login via a login script containing this command: `PS1='$PWD $ '`

*That makes your prompt string be the value of the PWD variable followed by a dollar sign followed by a space. The PWD variable always shows where you are in the UNIX file tree.*

## 2) Parse - the shell parses what you entered

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

*The shell parses what you entered and identifies the command, the options, the arguments and any redirection*

```
/home/cis90/simben/Poems/Yeats $ file *
```

2) Parse - the shell parses what you entered

**Change to your Poems/Yeats directory and parse this command:**

**file \***

Command:

Options:

Number of arguments:

Arguments:

Redirection:

*Put your answers in the chat window*

2) Parse - the shell parses what you entered

**Change to your Poems/Yeats directory and  
parse this command:**

**file \***

Command: file

Options: na

Number of arguments: 3

Arguments: mooncat old whitebirds

Redirection: na

## 2) Parse - the shell parses what you entered

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

**file \***

*How many arguments is this?*

*Use the **echo** command to find out*

```
/home/cis90/simben/Poems/Yeats $ echo *  
mooncat old whitebirds
```

*There are actually three arguments!*

### 3) Search - the shell searches the path for your command

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

```
/home/cis90/simben/Poems/Yeats $ file *
```

*The shell searches your path for the command you entered.*

*Where does the shell find the command?*

*You can do this yourself manually as follows:*

```
/home/cis90/simben/Poems/Yeats $ type file  
file is /usr/bin/file
```

*The **file** command is the **/usr/bin** directory.*

```
/home/cis90/simben/Poems/Yeats $ echo $PATH  
/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:.
```

*The **/usr/bin** directory is the 4<sup>th</sup> directory on your path. Each directory is delimited by ":" characters.*

## 4) Execute - the shell executes the command program file

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

```
/home/cis90/simben/Poems/Yeats $ file *
```

The next step is to load the **file** command that was found into memory. The program on the hard drive becomes a **process** in memory with a unique PID (Process ID).

Each new process is given three file descriptors **stdin**, **stdout** and **stderr** for input and output purposes.

These are sometimes referred to as the three **standard IO (Input/Output) streams**.

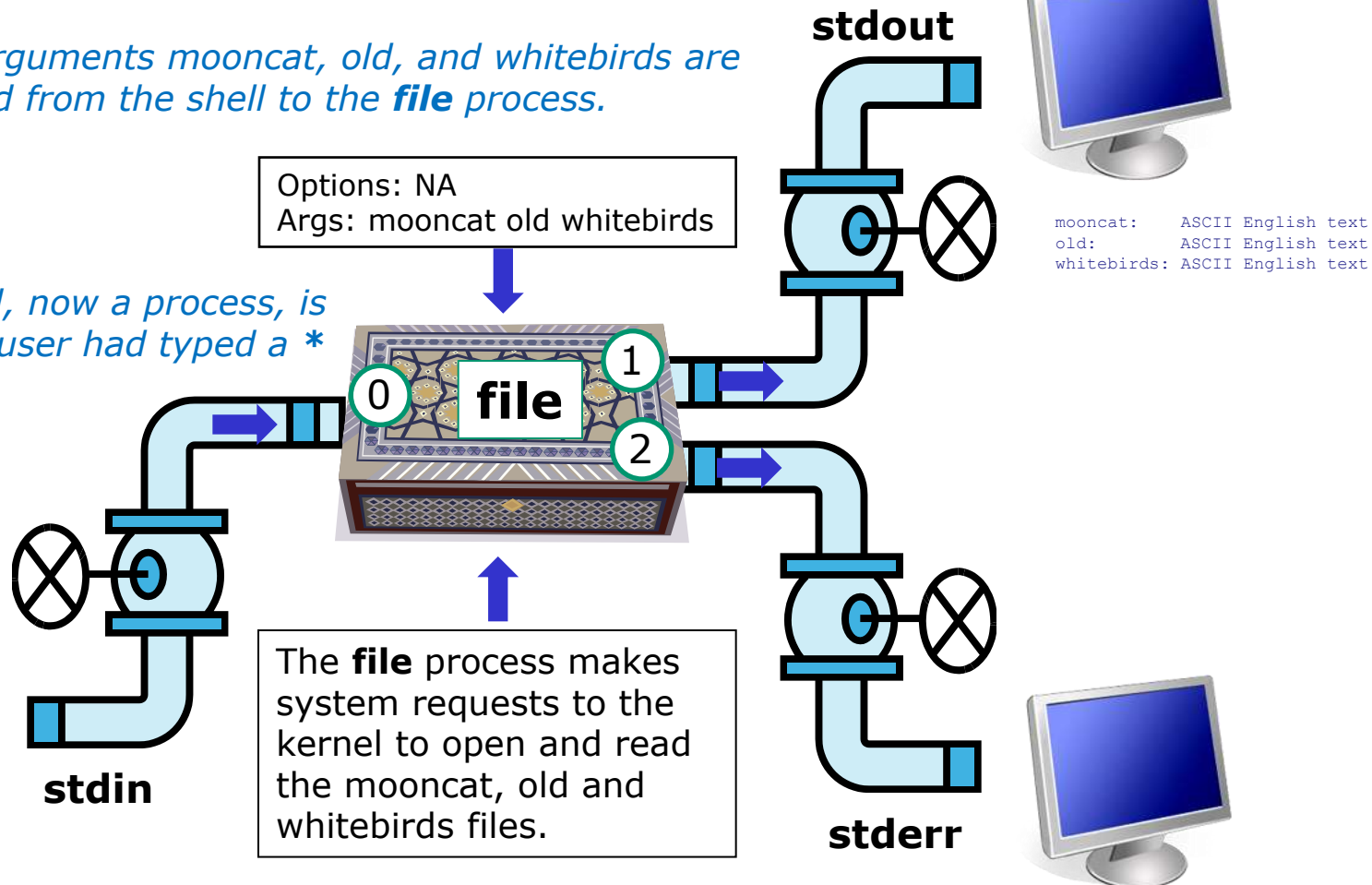
## 4) Execute - the shell executes the command program file

```
/home/cis90/simben/Poems/Yeats $ file *
```

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

*The arguments mooncat, old, and whitebirds are passed from the shell to the **file** process.*

*The **file** command, now a process, is unaware that the user had typed a \**





## 5) Nap - the shell sleeps while the command runs

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

```
/home/cis90/simben/Poems/Yeats $ file *
mooncat:      ASCII English text
old:          ASCII English text
whitebirds:   ASCII English text
```

} *Output from the file command*

*The shell's nap ends when the file command has finished*

## 6) Repeat - the shell does it again

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

*And then it does it all over again for the next command*



# Trouble on the island today



*Reminder to instructor:*

*On Sun-Hwa-vi, run trouble-L5 as root, rm /etc/nologin*

*Can you cat a file?*

## Warm-up Activity

From Opus, login to Sun-Hwa-L5 as follows:

**ssh sun-hwa-L5**

or **ssh \$LOGNAME@sun-hwa-L5**

or **ssh \$LOGNAME@sun-hwa-L5.cis.cabrillo.edu**

After logging in, try to **cat** this file: */etc/mensaje*

If successful:

then

click green "yes" check on CCC Confer  
Help your neighbor

else

Start TROUBLESHOOTING!



*Hint: Lesson 2*

# Housekeeping





- Lab 4 is due by 11:59PM tonight
- Use the **submit** command to turn in Lab 4
- There is a **check4** script available

## Fine print

- Test 1 is next week!

Test #1 is next week

Practice test available now





**Test #1 is next  
week**

**Practice test  
available now**



**Test #1 is next week**

**Practice test  
available now**

## Test next week

### **30 points, plus some extra credit:**

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

**Practice test systems shutdown before real test starts!**

## **Use the forum to discuss practice test questions**

- Post if you get stuck on a question
- Post to share tips with others
- Post if you would like a clarification on a test question
- Post your answer to a question to discuss and get feedback from others

## Use the forum to arrange study groups

### Practice Test 2 Study Group

Locked Search this topic...

3 posts • Page 1 of 1

#### Practice Test 2 Study Group

by Tess Pritchard • Wed Apr 01, 2015 11:59 am

I know it's late notice, but Mario and I are going to start working through the practice test tomorrow.  
Thursdays 1pm in the CIS Lab.  
We'd love it if you could join!

Thanks,  
Tess

Tess Pritchard

Posts: 30  
Joined: Wed Sep 10, 2014  
2:15 pm

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

#### online study group

by Benjamin Correia • Mon Mar 02, 2015 4:22 pm

I wasn't able to make the study group on campus today due to work so I thought I'd see if anyone would be willing to meet up online through Skype, hangouts or some other online collaborative work space like Docs...

let me know if your interested in spending an hour or two studying tomorrow morning from 9am to 11, I have work from 12pm until around 10:30 so if people would rather work a night I could devote an hour or so after that time.

if you have any ideas for a online study group pleas feel free to add your thoughts, I am having a bit of trouble really understanding a few things we covered in class so I thought I'd ask my peers for some advice if possible.

-Benji

Benjamin Correia

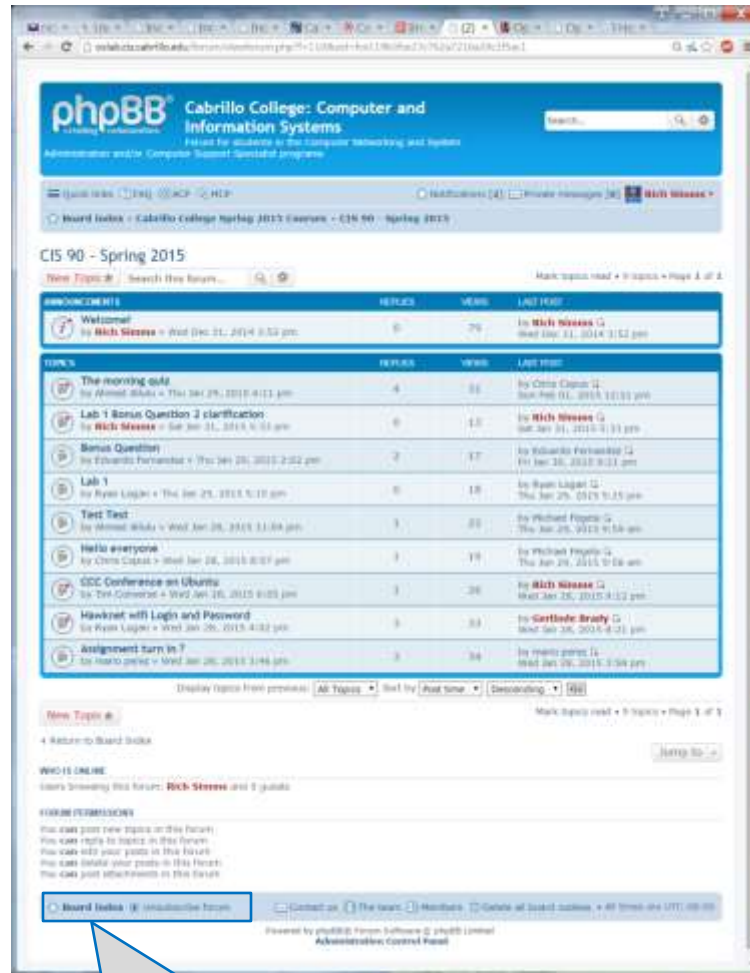
Posts: 28  
Joined: Tue Feb 03, 2015  
11:11 am

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

## Don't miss replies to your forum posts

2) Go to the CIS 90 forum

1) Login to the forum



3) Click the "Subscribe" link at the bottom so that it changes to "Unsubscribe".

Board index Unsubscribe forum

## Would you like some help learning Linux?



*If you would like some additional come over to the CIS Lab. There are student lab assistants and instructors there to help you.*

*Tess, Michael, and Paul are  
CIS 90 Alumni.*

*Mike Matera is the other  
Linux instructor.*

*I'm in there Mondays  
11:00-1:30pm.*



# Don't Forget -- Perkins/VTEA Survey

The screenshot shows a forum post on the 'Cabrillo College: Computer and Information Systems' board. The post is titled 'Carl D. Perkins Vocational and Technical Education Act' and is authored by 'Rich Stevens'. The post text explains that the act was originally authorized by Congress in 1964, reauthorized in 1998, and again in 2006. It provides federal funding for improving career technical education (CTE) within the United States. The post instructs students to complete a survey to receive a portion of this funding. The survey can be completed online using a web address: <http://www.cabrillo.edu>. The post also provides instructions on how to log on to WEBADVISOR, select 'STUDENTS: Click Here', and answer questions by clicking on the circle to the left of your 'Yes' or 'No' answers. The post concludes with 'Then "LOG OUT"' and a thank you message.

*This is an important source of funding for Cabrillo College.*

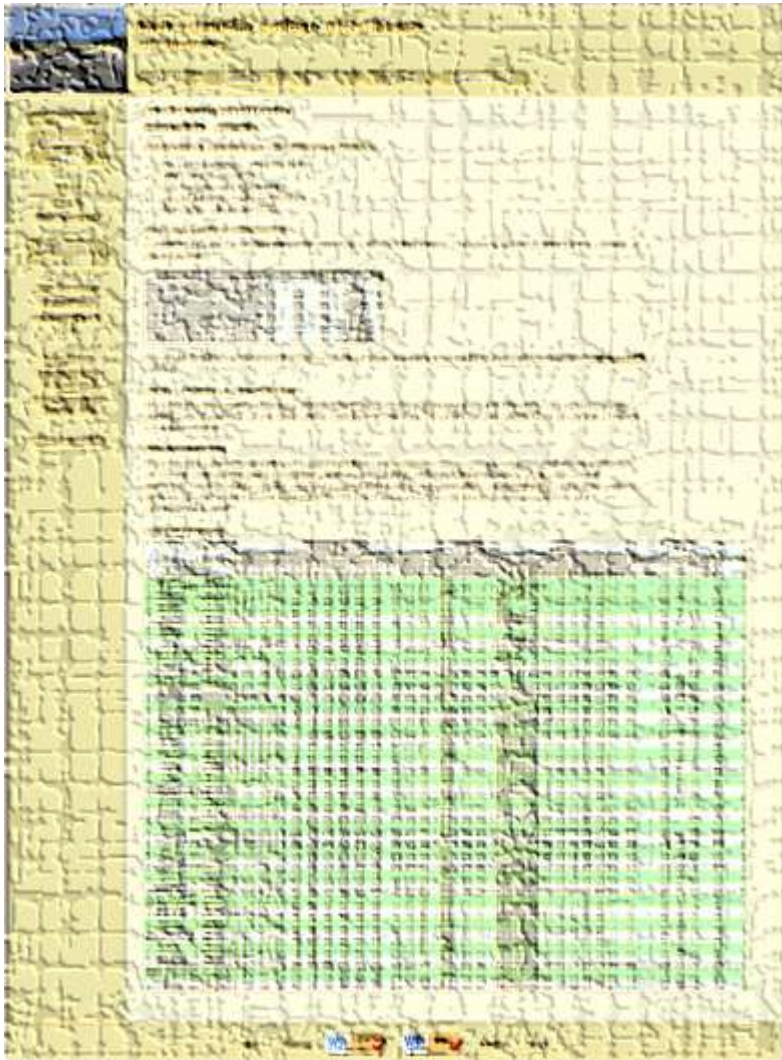
*Send me an email stating you completed this survey for **three points extra credit!***

The screenshot shows a 'Career Technical Information' survey form. The form is titled 'Career Technical Information' and includes the instruction: 'Your answers to these questions will help qualify Cabrillo College for Perkins/VTEA grant funds.' The form contains several questions, each with a 'Yes' or 'No' radio button. The questions are: 'Are you currently receiving benefits from:', 'TAFICALWORKS:', 'UI (Supplemental Security Income)', 'GA (General Assistance)', 'Does your [SSN](#) qualify you for a tax waiver?', 'Are you a single parent with custody of one or more minor children?', 'Are you a [dependent care](#) allowing Cabrillo to receive job aids?', and 'Have you moved in the preceding 30 months to attend, or to accompany parents or spouses to attend, temporary or seasonal employment or agricultural, dairy, or fishing?'.

<http://oslab.cis.cabrillo.edu/forum/viewtopic.php?f=114&t=3863>



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



# GRADES

*Be sure and check your progress on the Grades page as the course continues on.*

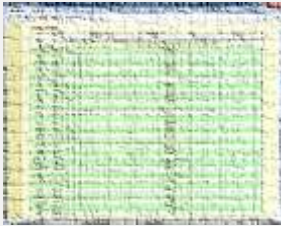
*Send me a student survey if you haven't already to get your LOR secret code name.*

## Where to find your grades

*Send me your survey to get your LOR code name.*

### The CIS 90 website Grades page

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



### Points that could have been earned to date:

3 quizzes: 9 points  
 3 labs: 90 points  
 1 forum quarter: 20 points  
**Total: 119 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

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

### Or check on Opus

**checkgrades** *codename*  
 (where *codename* is your LOR codename)



Written by Jesse Warren a past CIS 90 Alumnus

**grades** *codename*  
 (where *codename* is your LOR codename)



Written by Sam Tindell a past CIS 90 Alumnus.  
 Try his tips, schedule and forums scripts as well!

## LPI Linux Essentials Certificate

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

Objective	Description	Click to Access
000	What is LPI Linux Essentials	<a href="#">Click to Access</a>
1.1	Linux evolution and popular operating systems	<a href="#">Click to Access</a>
1.2	Major Open Source applications	<a href="#">Click to Access</a>
1.3	Understanding Open Source Software and licensing	<a href="#">Click to Access</a>
1.4	ECT skills and working with Linux	<a href="#">Click to Access</a>
2.1	Command line basics	<a href="#">Click to Access</a>
2.2	Using the command line to get help	<a href="#">Click to Access</a>
2.3	Using directories and listing files	<a href="#">Click to Access</a>
2.4	Creating, moving and deleting	<a href="#">Click to Access</a>
3.1	Archiving files from the command line	<a href="#">Click to Access</a>
3.2	Backups and extracting data from files	<a href="#">Click to Access</a>
3.3	Turning commands into a script	<a href="#">Click to Access</a>
4.1	Choosing an operating system	<a href="#">Click to Access</a>
4.2	Understanding computer hardware	<a href="#">Click to Access</a>
4.3	Where data is stored	<a href="#">Click to Access</a>
4.4	Your computer on the network	<a href="#">Click to Access</a>
5.1	Basic security and user types	<a href="#">Click to Access</a>
5.2	Creating users and groups	<a href="#">Click to Access</a>
5.3	Manage file permissions and ownership	<a href="#">Click to Access</a>
5.4	Special directories and files	<a href="#">Click to Access</a>

Inductive and live video based Linux Training

<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

**2.3 Major Open Source Applications**

The Linux kernel is not a wide variety of software services mostly hardware platforms. A computer can act as a server, which means it primarily handles data on other's behalf, or as a desktop, which means a user will be interacting with it directly. The machine can run software or it can be used as a development machine in the process of creating software. You can even use multiple roles as there is no restriction to Linux about the role of the machine, it's mainly a matter of configuring which applications run.

One advantage of this is that you can simulate almost all aspects of a production environment from developing, to testing, to deployment on actual server hardware, which saves costs and time. As someone learning Linux, you can run the same server applications on your desktop or inexpensive virtual server that you run on a large internet Service Provider. Of course, you will not be able to handle the volume a large provider results, as they will have much more expensive hardware, but you can simulate almost any configuration without needing powerful hardware or server licensing.

Linux software generally falls into one of three categories:

- Server software** - software that has no direct interaction with the monitor and keyboard of the machine it runs on. Its purpose is to serve information to other computers, called *clients*. Sometimes server software may not talk to other computers but will get all these and "force" it data.
- Desktop software** - a web browser, text editor, music player, or other software that you interact with. In many cases, such as a web browser, the software is talking to a server on the other end and downloading the data for you. While the desktop software is the client.
- Tools** - a broad category of software that tends to assist a user to manage your system. You might have a tool that helps you configure your display, or something that provides a Linux shell, or even more sophisticated tools that convert entire code to something that the computer can read.

Additional, we will consider mobile applications, available for the kernel of the Linux.

Test Export...zip

<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.*



Everything  
is a file

## Everything is a file in UNIX (even a terminal)

- A terminal
- A file
- A directory
- A hard drive
- A hard drive partition
- A CD
- A partition on a USB flash drive
- Kernel run-time information



*Implemented as  
files in UNIX*

## Everything is a file in UNIX (even a terminal)

- A terminal *e.g. /dev/pts/2*
- A file *e.g. /home/cis90/simben/letter*
- A directory *e.g. /home/cis90/*
- A hard drive *e.g. /dev/sda*
- A hard drive partition *e.g. /dev/sda1*
- A CD *e.g. /dev/cdrom*
- A partition on a USB flash drive *e.g. /dev/sdb2*
- Kernel run-time information *e.g. /proc/sys/kernel/hostname*

# Everything is a file in UNIX (even a terminal)

```
[rsimms@oslab ~]$ ls -l /dev/pts/3
crw--w----. 1 leebri90 tty 136, 3 Sep 30 16:33 /dev/pts/3
```

*terminal*

```
[rsimms@oslab ~]$ ls -l /home/cis90/simben/letter
-rw-r--r--. 1 simben90 cis90 1044 Jul 20 2001 /home/cis90/simben/letter
```

*file*

```
[rsimms@oslab ~]$ ls -ld /home/cis90/
drwxr-x---. 43 rsimms cis90 4096 Sep 16 15:00 /home/cis90/
```

*directory*

```
[rsimms@oslab ~]$ ls -l /dev/sda
brw-rw----. 1 root disk 8, 0 Sep 13 17:47 /dev/sda
```

*hard drive*

```
[rsimms@oslab ~]$ ls -l /dev/sda1
brw-rw----. 1 root disk 8, 1 Sep 13 17:47 /dev/sda1
```

*partition*

```
[rsimms@oslab ~]$ ls -l /dev/cdrom
lrwxrwxrwx. 1 root root 3 Sep 13 17:46 /dev/cdrom -> sr0
```

```
[rsimms@oslab ~]$ ls -l /dev/sr0
brw-rw----. 1 root cdrom 11, 0 Sep 13 17:46 /dev/sr0
```

*CD drive*

```
[rsimms@oslab ~]$ ls -l /proc/sys/kernel/hostname
-rw-r--r-- 1 root root 0 Sep 24 15:45 /proc/sys/kernel/hostname
```

*Kernel runtime info*



# File Types

Long listing code (ls -l)	Type	How to make one
d	directory	mkdir
-	Regular file <ul style="list-style-type: none"> <li>• Programs</li> <li>• Text</li> <li>• Data (binary)</li> </ul>	touch
l	symbolic link	ln -s
c	character device file	mknod
b	block device file	mknod

Common file types in a Linux ext<sub>n</sub> file system

# Everything is a file in UNIX (even a terminal)

## Nice things about files

- you can write to them

```
[rsimms@opus ~]$ echo "Rich was here" > myfile
```

- and read from them

```
[rsimms@opus ~]$ cat myfile  
Rich was here
```

## Class Activity

- Write to a file

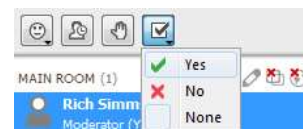
```
echo "Rumpelstiltskin was here" > myfile
```

- Read the file

```
cat myfile
```

Did it work?

Click green "Yes" check or red "No" X  
on CCC Confer



## Everything is a file (even a terminal)

```
/home/cis90/simmsben $ tty  
/dev/pts/1
```

*Use the **tty** command to identify the specific terminal device being used*

*Note this device is identified using an absolute pathname*

# Everything is a file (even a terminal)

```
/home/cis90/simmsben $ tty  
/dev/pts/1
```

*Show which terminal you are using*

```
/home/cis90/simmsben $ who  
simmsben pts/1      2010-09-29 07:38 (dsl-49-64-10-90.dhcp.cruzio.com)  
srecklau pts/2      2010-09-29 06:06 (62.143.60.194)  
rsimms pts/4        2010-09-29 06:47 (dsl-49-64-10-90.dhcp.cruzio.com)
```

*Use who to see who is logged in*

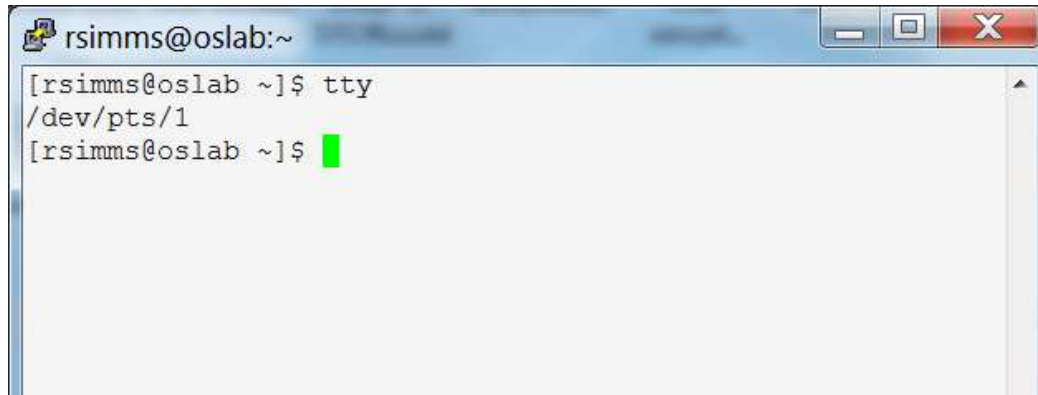
```
/home/cis90/simmsben $ ls -l /dev/pts/*  
crw--w---- 1 simmsben tty 136, 1 Sep 29 07:45 /dev/pts/1  
crw--w---- 1 srecklau tty 136, 2 Sep 29 07:44 /dev/pts/2  
crw--w---- 1 rsimms tty 136, 4 Sep 29 06:48 /dev/pts/4
```

*Do a long listing to see  
all the terminal devices  
in use*

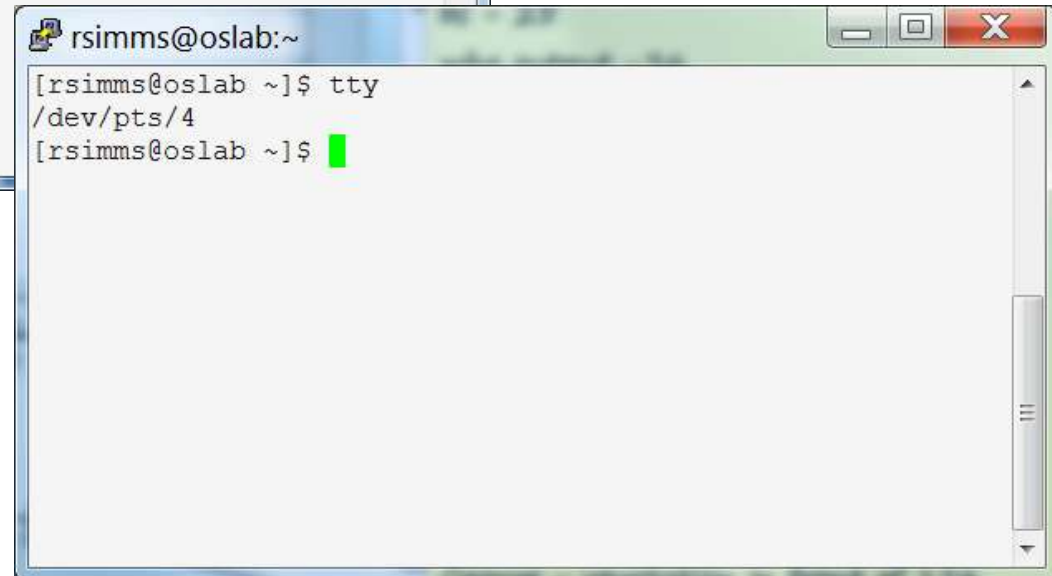
*Notice the owner is someone who has logged in*

*Notice the file type is "c" which is a character device file*

# Everything is a file (even a terminal)



```
rsimms@oslab:~  
[rsimms@oslab ~]$ tty  
/dev/pts/1  
[rsimms@oslab ~]$ █
```

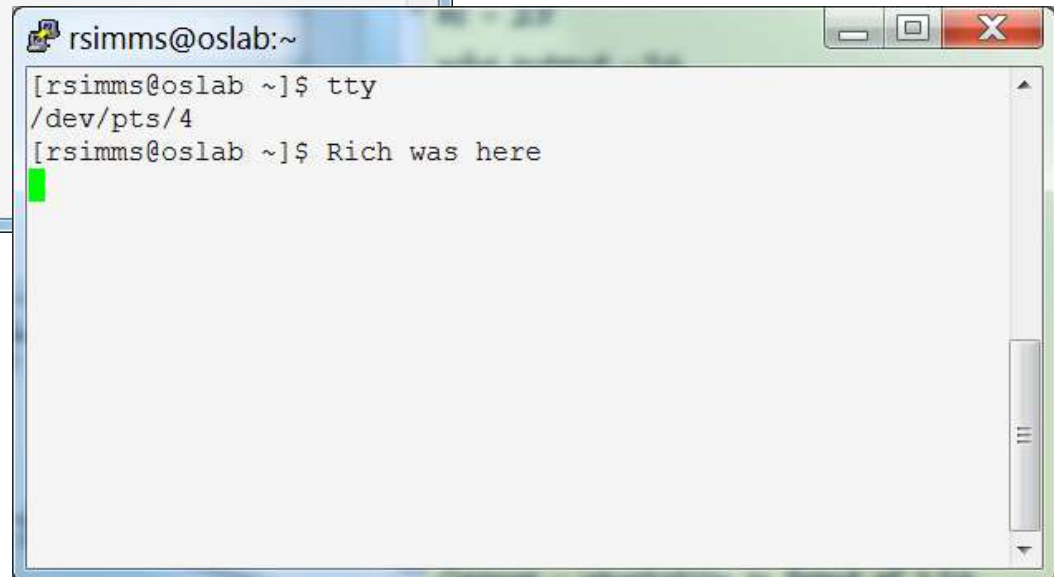


```
rsimms@oslab:~  
[rsimms@oslab ~]$ tty  
/dev/pts/4  
[rsimms@oslab ~]$ █
```

# Everything is a file (even a terminal)



```
rsimms@oslab:~  
[rsimms@oslab ~]$ tty  
/dev/pts/1  
[rsimms@oslab ~]$ echo "Rich was here" > /dev/pts/4  
[rsimms@oslab ~]$ █
```



```
rsimms@oslab:~  
[rsimms@oslab ~]$ tty  
/dev/pts/4  
[rsimms@oslab ~]$ Rich was here  
█
```

# Everything is a file (even a terminal)

```

rsimms@oslab:~
[rsimms@oslab ~]$ tty
/dev/pts/1
[rsimms@oslab ~]$ echo "Rich was here" > /dev/pts/4
[rsimms@oslab ~]$ echo "Rich was here" > myfile
[rsimms@oslab ~]$
[rsimms@oslab ~]$ ls -l /dev/pts/4 myfile
crw-----. 1 rsimms tty    136, 4 Sep 30 20:23 /dev/pts/4
-rw-r--r--. 1 rsimms staff  14 Sep 30 20:25 myfile
[rsimms@oslab ~]$

```

*absolute path* (points to /dev/pts/4)

*relative path* (points to myfile)

*do long listing* (points to ls -l)

*regular file type* (points to -rw-r--r--)

*character device file type* (points to crw)

*a regular file* (points to myfile)

*a terminal device* (points to /dev/pts/4)



## Class Activity

### Part I

- Login into Opus
- Use **echo "I can do it" > myfile**
- Print your new file with **cat myfile**

```
simben90@oslab:~$ echo "I can do it" > myfile
simben90@oslab:~$ cat myfile
I can do it
simben90@oslab:~$ tty
/dev/pts/5
simben90@oslab:~$ cat myfile > /dev/pts/9
simben90@oslab:~$
```

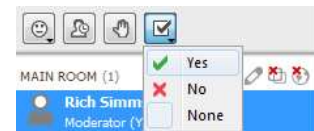
### Part II

- Open a second session on Opus
- You should have two terminals now
- In both sessions use **tty** to identify the terminal devices being used
- In one terminal use **cat myfile > /dev/pts/xx** where xx is your other terminal

```
simben90@oslab:~$ tty
/dev/pts/9
simben90@oslab:~$ I can do it
simben90@oslab:~$
```

Did it work?

Click green "Yes" check or red "No" X  
on CCC Confer



# More File Name Expansion

(also known as *globbing*)

# Filename Expansion Metacharacters

**\***

matches all non-hidden filenames in the current directory when used alone or zero or more characters when used as a prefix, infix or postfix.

**?**

matches any single character in any of your current directory's filenames.

**[]**

matches any single character contained within the brackets.



*You may also hear this process called "globbing"*

## Shell Parse Step

*Filename expansion happens during the shell parsing step, before the command is even located or executed.*

- 1) Prompt
- 2) Parse**
- 3) Search for program (along the path)
- 4) Execute program
- 5) Nap (wait till process is done)
- 6) Repeat

*The commands never see \*, ?, and [] expansion characters. These characters get replaced by the shell before the command is even located and executed.*

## The \* Filename Expansion Metacharacter

Example: **text.\*** will be expanded by the shell to match any files that start with "text."

```
/home/cis90/simben $ echo text.*
text.err text.fxd
```

```
/home/cis90/simben $ ls -i text.*
19496 text.err 19497 text.fxd
```

```
/home/cis90/simben $ file text.*
text.err: ASCII text
text.fxd: ASCII text
```

```
/home/cis90/simben $ wc -l /home/cis90/mahtab/text.*
 11 /home/cis90/mahtab/text.err
 10 /home/cis90/mahtab/text.fxd
 21 total
```

```
/home/cis90/simben $ tail -n1 ../mahtab/text.*
==> ../mahtab/text.err <==
number10.
```

```
==> ../mahtab/text.fxd <==
This is line number 10.
```

## The \* Filename Expansion Metacharacter

Example: \* is expanded to match all directories in /home/cis90 and **ti\*** to match all files starting with "ti"

```
/home/cis90/simben $ ls -l ../*/Poems/Blake/ti*
-rw-r--r--. 1 beakie90 cis90 115 Jul 20 2001 ../beakie/Poems/Blake/tiger
-rw-r--r--. 1 calmic90 cis90 115 Jul 20 2001 ../calmic/Poems/Blake/tiger
-rw-r--r--. 1 casenr90 cis90 115 Jul 20 2001 ../casenr/Poems/Blake/tiger
-rw-r--r--. 1 casric90 cis90 115 Jul 20 2001 ../casric/Poems/Blake/tiger
-rw-r--r--. 1 cis90     cis90 115 Jul 20 2001 ../cis/Poems/Blake/tiger
-rw-r--r--. 1 daweli90 cis90 115 Jul 20 2001 ../daweli/Poems/Blake/tiger
-rw-r--r--. 1 fahmic90 cis90 115 Jul 20 2001 ../fahmic/Poems/Blake/tiger
-rw-r--r--. 1 fitcon90 cis90 115 Jul 20 2001 ../fitcon/Poems/Blake/tiger
< snipped >
-rw-r--r--. 1 simben90 cis90 115 Jul 20 2001 ../simben/Poems/Blake/tiger
-rw-r--r--. 1 specod90 cis90 115 Jul 20 2001 ../specod/Poems/Blake/tiger
-rw-r--r--. 1 thinic90 cis90 115 Jul 20 2001 ../thinic/Poems/Blake/tiger
-rw-r--r--. 1 tilbuz90 cis90 115 Jul 20 2001 ../tilbuz/Poems/Blake/tiger
-rw-r--r--. 1 vasjor90 cis90 115 Jul 20 2001 ../vasjor/Poems/Blake/tiger
-rw-r--r--. 1 vivrut90 cis90 115 Jul 20 2001 ../vivrut/Poems/Blake/tiger
-rw-r--r--. 1 weljon90 cis90 115 Jul 20 2001 ../weljon/Poems/Blake/tiger
-rw-r--r--. 1 weltim90 cis90 115 Jul 20 2001 ../weltim/Poems/Blake/tiger
/home/cis90/simben $
```

## The \* Filename Expansion Metacharacter

Note, DOS uses \*.\* to match all files.

BUT, this is NOT true in UNIX

```
/home/cis90/simmsben $ echo *.*  
Lab2.0 Lab2.1 text.err text.fxd
```

*Instead, \*.\* is expanded to match all files in the current directory containing a "."*

## The \* Filename Expansion Metacharacter

*Note the \* metacharacter by itself does not match any hidden files in your current working directory*

```
/home/cis90/simmsben $ echo *
bigfile bin delete empty Hidden Lab2.0 Lab2.1 letter Miscellaneous mission
Poems proposal1 proposal2 proposal3 small_town spellk text.err text.fxd
timecal what_am_i
```

```
/home/cis90/simmsben $ ls -a
.          .bashrc  empty    letter    Poems     spellk
.zshrc
..         bigfile  Hidden   Miscellaneou proposal1  text.err
.bash_history bin      Lab2.0   mission   proposal2  text.fxd
.bash_logout delete   Lab2.1   .mozilla  proposal3  timecal
.bash_profile .emacs  .lessht  .plan     small_town what_am_i
```



## The ? Filename Expansion Metacharacter

Example: **???** will be match any three character file name

```
/home/cis90/simben $ echo ???
bin log
```

```
/home/cis90/simben $ ls bin/???
bin/app
```

```
/home/cis90/simben $ ls /bin/???
/bin/awk /bin/csh /bin/env /bin/pwd /bin/red /bin/rvi /bin/tar
/bin/cat /bin/cut /bin/ksh /bin/raw /bin/rpm /bin/sed
```

```
/home/cis90/simben $ tail -n3 /home/cis90/grodav/bin/???
while :
do sleep 1
done
```

```
/home/cis90/simben $ file /home/cis90/grodav/Poems/*/???
/home/cis90/grodav/Poems/Yeats/old: ASCII English text
```

## The [] Filename Expansion Metacharacter

Example: **[12]** will match a "1" or a "2"

```
/home/cis90/simben $ head -n1 Poems/Shakespeare/sonnet[12]
```

```
==> Poems/Shakespeare/sonnet1 <==
```

```
From fairest creatures we desire increase,
```

```
==> Poems/Shakespeare/sonnet2 <==
```

```
When forty winters shall besiege thy brow,
```

```
/home/cis90/simben $ wc ../balcor/Poems/Shakespeare/sonnet[12]
```

```
14 105 614 ../balcor/Poems/Shakespeare/sonnet1
```

```
14 114 631 ../balcor/Poems/Shakespeare/sonnet2
```

```
28 219 1245 total
```

```
/home/cis90/simben $ ls -d /etc/*[12]*
```

```
/etc/dbus-1 /etc/iproute2 /etc/pnm2ppa.conf /etc/rc2.d
```

```
/etc/DIR_COLORS.256color /etc/mke2fs.conf /etc/polkit-1 /etc/sasl2
```

```
/etc/gtk-2.0 /etc/pbm2ppa.conf /etc/rc1.d /etc/X11
```



# File Name Expansion (also known as globbing)

## PRACTICE QUESTIONS

## The \* Filename Expansion Metacharacter

*Your turn now*

What command would classify all files in the parent directory that start with m?

*Write your answer in the chat window*

## The \* Filename Expansion Metacharacter

### Answer

What command would classify all files in the parent directory that start with m?

```
/home/cis90/simben $ file ../m*  
../mcgcam: directory  
../milhom: directory  
/home/cis90/simben $
```

*Note, the matches can change each term since it is based on student names.*

## The ? Filename Expansion Metacharacter



*A ? matches exactly one character which could be anything*

What command would list all 13 character filenames in /bin

*Write your answer in the chat window*

## The ? Filename Expansion Metacharacter

What command would list all 13 character filenames in /bin

*Answer*

```
/home/cis90/simben $ ls /bin/??????????????  
/bin/dnsdomainname /bin/nisdomainname /bin/unicode_start
```

## The [] Filename Expansion Metacharacter



*A [] will match any character between the brackets*

From your home directory, what command would print the first line of all Shakespeare sonnets ending in a 2 or 5?

*Write your answer in the chat window*



## The [] Filename Expansion Metacharacter

From your home directory, what command would print the first line of all Shakespeare sonnets ending in a 2 or 5?

### *Answer*

```
/home/cis90/simben $ head -n 1 Poems/Shakespeare/*[25]
==> Poems/Shakespeare/sonnet15 <==
When I consider every thing that grows

==> Poems/Shakespeare/sonnet2 <==
When forty winters shall besiege thy brow,

==> Poems/Shakespeare/sonnet35 <==
Whoever hath her wish, thou hast thy Will,

==> Poems/Shakespeare/sonnet5 <==
Those hours that with gentle work did frame
/home/cis90/simben $
```

## Filename Expansion Metacharacters

\* ? []

What commands are there in /usr/bin that start with a "n" or "m", are 5 letters long and end with a "p"?

*Hint: Use a combination of filename expansion metacharacters*

*Write your answer in the chat window*

## Filename Expansion Metacharacters

\* ? []

What commands are there in /usr/bin that start with a "n" or "m", are 5 letters long and end with a "p"?

*Answer*

```
/home/cis90/simben $ echo /usr/bin/[nm]??p  
/usr/bin/nohup
```

## Filename Expansion Metacharacters

\* ? []

For the command:

```
file /usr/share/man/*/ [ap]?? .8.gz
```

What arguments are actually getting passed to the **file** command to process?

## Filename Expansion Metacharacters

\* ? []

For the command:

```
file /usr/share/man/*/ [ap]?? .8.gz
```

What arguments are actually getting passed to the **file** command to process?

```
/home/cis90/simben $ echo /usr/share/man/*/ [ap]?? .8.gz  
/usr/share/man/man8/arp.8.gz /usr/share/man/man8/atd.8.gz  
/usr/share/man/man8/pam.8.gz /usr/share/man/man8/pvs.8.gz
```

*Tip: Use echo to expand complicated filenames containing multiple filename expansion characters*



# Command Review

*Use the **man** command or google for the details*

New commands:

cal	- show calendars
clear	- clear the terminal screen
exit	- terminate your shell and log off
history	- show previous commands
hostname	- show the name of the computer being accessed
id	- show user and group id information
ps	- show processes (loaded programs) being run
ssh	- secure login to a remote system
uname	- shows kernel information
tty	- show terminal information
who	- show who else is logged on
who am i	- Identifies which login session you are using
Ctrl-Alt-F1 to Ctrl-Alt-F7	- Change between terminals and X windows (graphics)

New Files and Directories:

VMware:

Ctrl-Alt	- to move mouse cursor out of VM
----------	----------------------------------

Use the **man** command or google for the details

New commands:

- |         |  |
|---------|--|
| apropos | - search for string in whatis database |
| bc      | - binary calculator                    |
| cat     | - print file(s)                        |
| cd      | - change directory                     |
| echo    | - print text                           |
| env     | - show shell environment variables     |
| info    | - online documentation with hot links  |
| file    | - show file information                |
| ls      | - show directory contents              |
| passwd  | - change password                      |
| set     | - show (or set) shell variables        |
| type    | - show command location in path        |
| man     | - manual page for a command            |
| whatis  | - command summary                      |

New Files and Directories:

- |             |  |
|-------------|--|
| /etc/passwd | - user accounts  |
| /etc/shadow | - encrypted passwords                                  |
| /bin        | - directory of commands                                |
| /sbin       | - directory of superuser commands                      |
| /usr/bin    | - directory of commands, tools and utilities           |
| /usr/sbin   | - directory of superuser commands, tools and utilities |



New commands:

mail

?	print these commands
p <message list>	print messages
n	goto and print next message
e <message list>	edit messages
d <message list>	delete messages
s <message list> file	save (append) messages to file
u <message list>	undelete messages
R <message list>	reply to sender(s)
r <message list>	reply to all
m <user list>	mail to specific users
q	quit, saving read messages to local mbox file
x	quit, mark all mail as unread and undeleted.
h	print out active message headers

mesg

- Enable or disable writes to your terminal

write

- Write message to another user

New Files and Directories:

/var/mail

- Message store for mail

/var/mail/*username*

- Incoming mailbox for *username*

mbox

- File in users home directory where read messages are archived to

Use the **man** command or google for the details

Use the **man** command or google for the details

Commands:

cat	Print a file on the screen
cd	Change directory
file	Classify a file
head	View first several lines of a file
less	Scroll up and down long files
ls	List files
more	Scroll down long files
pwd	Print working directory
reset	Use to reset terminal window
tail	View last several lines of a file
wc	Count the words, lines or characters in a file
xxd	View (hex dump) binary/data files

New Files and Directories:

/	Root of the file tree
/home	Opus home directories
/home/cis90	CIS 90 class home directories
/home/cis90/ <i>username</i>	The home directory for CIS 90 student <i>username</i>

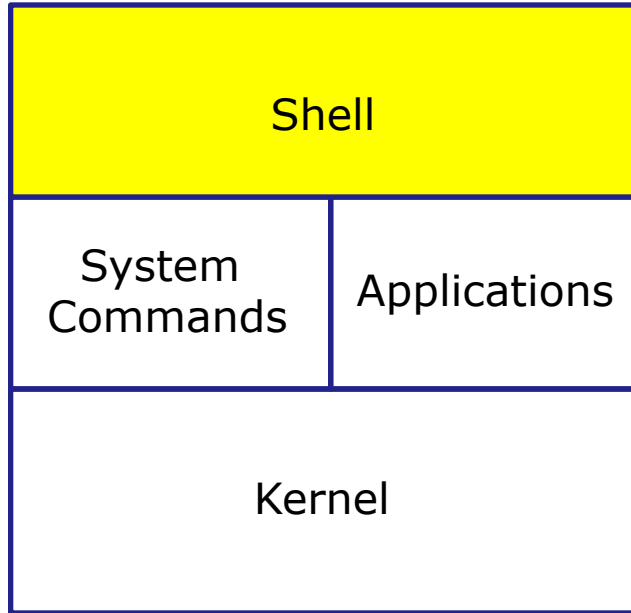


# Command line Syntax & Parsing

(review)



# Life of the Shell



- 1) **Prompt** for a command
- 2) **Parse** (interpret metacharacters, expand file names and dissect command line into options, arguments and redirection)
- 3) **Search** for program (along the path)
- 4) **Execute** program by loading into memory (becomes a process), hookup input and outputs, and pass along command line options and arguments.
- 5) **Nap** (wait till process is done)
- 6) **Repeat**

# Command Syntax

Command

Options

Arguments

Redirection

**Command** – is the name of an executable program file.

**Options** – various options which control how the program will operate.

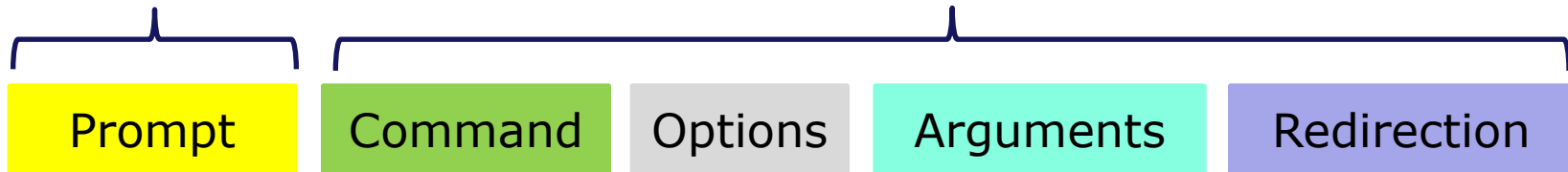
**Arguments** – the objects the command is directed to work upon.

**Redirection** – The default input stream (stdin) is from the console keyboard, the default output (stdout) and error (stderr) streams go to the console screen. Redirection can modify these streams to other files or devices.

# Command Syntax

Shell prints  
this to prompt  
user to enter a  
command

Shell parses this command line



## Examples

**Options** modify the  
behavior of the command

**Arguments** are what the  
command works upon

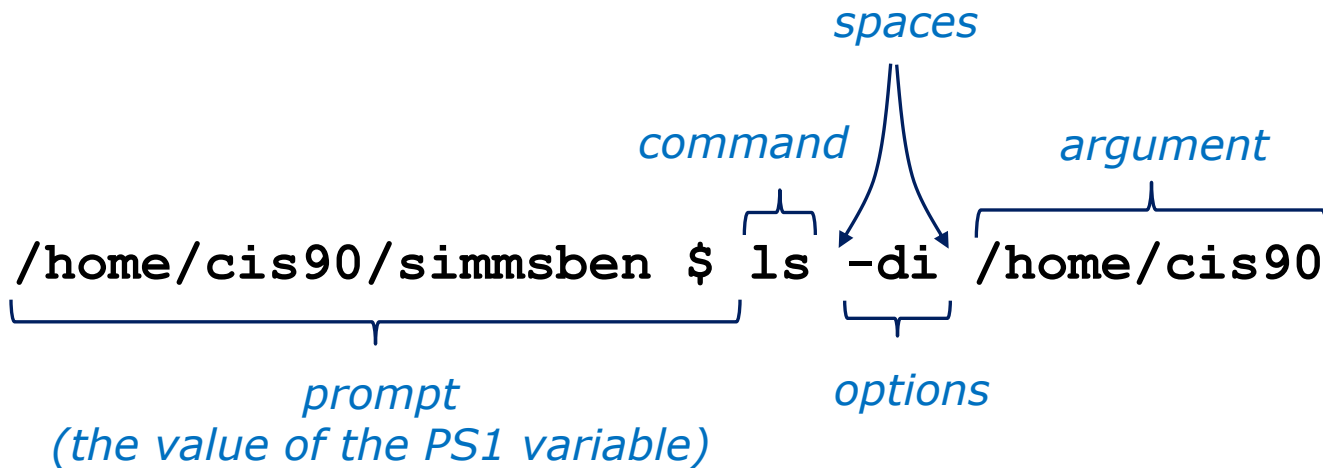
**Redirection** is covered  
later in the course

```

/home/cis90/simmsben $
/home/cis90/simmsben $ ls
/home/cis90/simmsben $ ls -l
/home/cis90/simmsben $ ls -lt
/home/cis90/simmsben $ ls -lt Poems/
/home/cis90/simmsben $ ls -lt Poems/ bin/
/home/cis90/simmsben $ ls -lt Poems/ bin/ > mylist
    
```

**Spaces (blanks)** are used to separate the command,  
options and arguments.

## Command Line Syntax Review



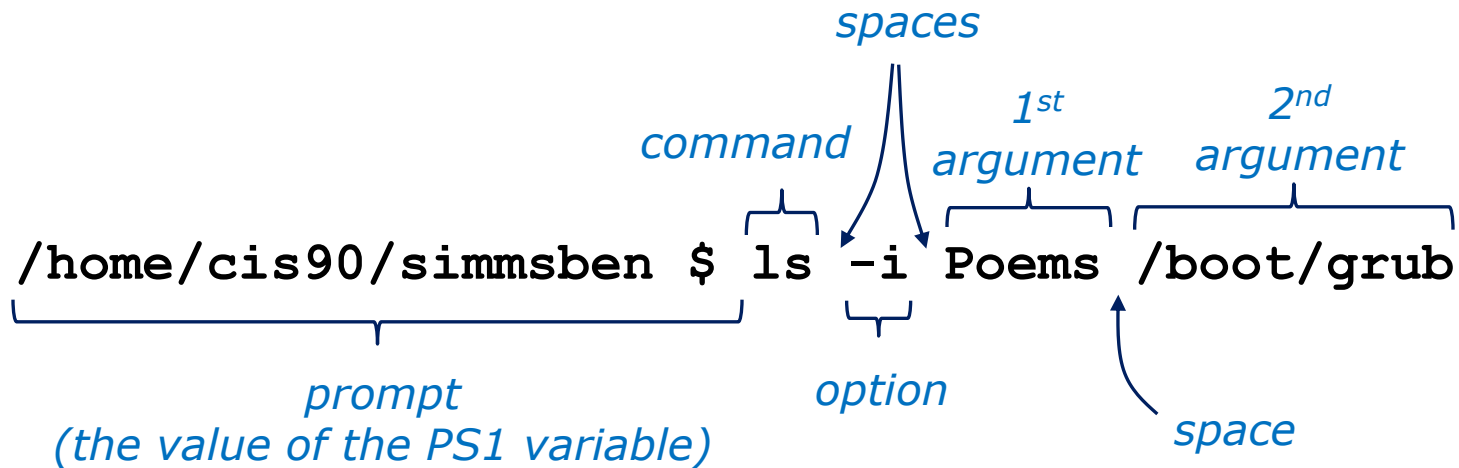
Parsing the command line above yields:

One command: **ls**

Two options: **d** and **i**

One argument: **/home/cis90** (an absolute pathname to a directory)

## Command Line Syntax Review



Parsing the command line above yields:

One command: **ls**

One options: **i**

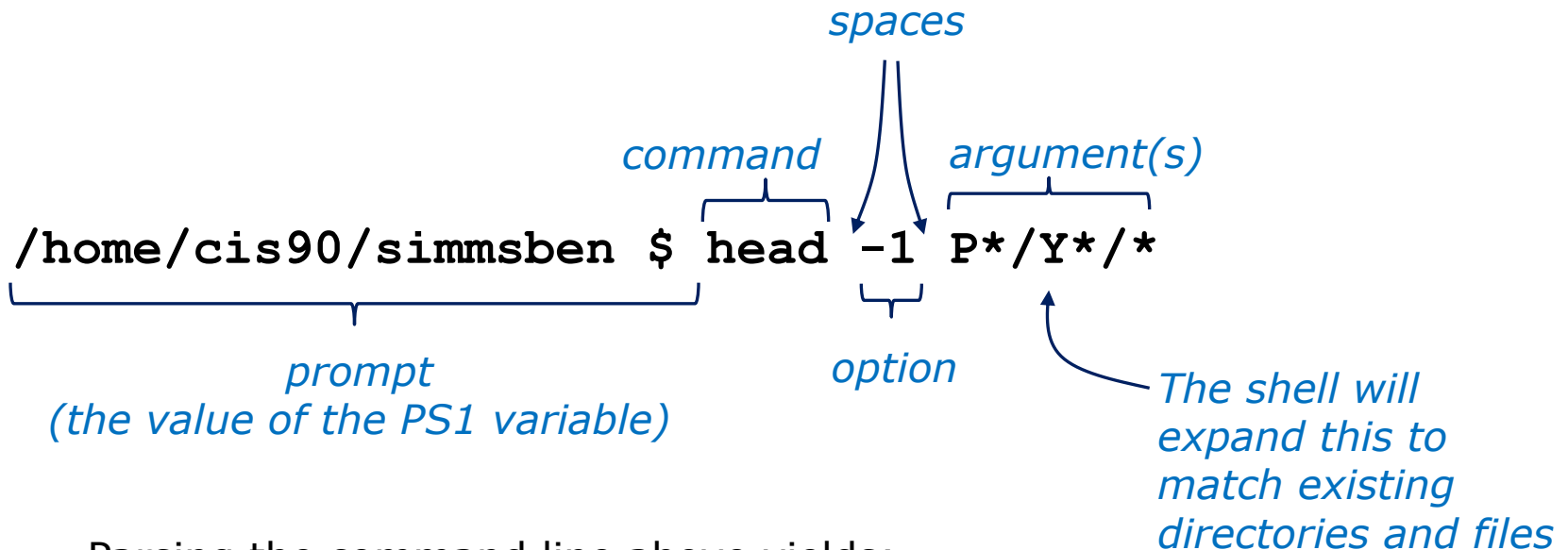
Two arguments:

**Poems** (a relative pathname to a directory)

**/boot/grub** (an absolute pathname to a directory)



# Command Line Syntax Review



Parsing the command line above yields:

One command: **head**

One option: **1**

Three arguments:

**Poems/Yeats/mooncat** (a relative pathname to a file)

**Poems/Yeats/old** (a relative pathname to a file)

**Poems/Yeats/whitebirds** (a relative pathname to a file)



# Command line Syntax & Parsing

## PRACTICE QUESTIONS

## Your turn now!

```
head -n1 /home/cis90/???t*/P*/Shake*/s????t[13]
```

Parse the command above and identify and see if you can identify:

1. The command
2. The options
3. The number of arguments:
4. The actual arguments:

## Your turn now!

```
head -n1 /home/cis90/???t*/P*/Shake*/s????t[13]
```

Parse the command above and identify:

1. The command: **head**

*The command is the head command*

## Your turn now!

```
head -n1 /home/cis90/???t*/P*/Shake*/s????t[13]
```

Parse the command above and identify:

1. The command: **head**
2. The options: **-n1**

*The options are -n1 which the user added to instruct the head command to output only one line*

## Your turn now!

```
head -n1 /home/cis90/???t*/P*/Shake*/s????t[13]
```

Parse the command above and identify:

1. The command: **head**
2. The options: **-n1**
3. The number of arguments: 4

*Note, the number of arguments can change each term since it is based on student names. How many are there today?*

*The number of arguments is not apparent without first processing all the filename expansion characters. You can use the echo command to see exactly how the expansion will be done.*

```
/home/cis90/simben $ echo /home/cis90/???t*/P*/Shake*/s????t[13]
/home/cis90/juetay/Poems/Shakespeare/sonnet1 /home/cis90/juetay/Poems/Shakespeare/sonnet3
/home/cis90/prites/Poems/Shakespeare/sonnet1 /home/cis90/prites/Poems/Shakespeare/sonnet3
/home/cis90/simben $
```

## Your turn now!

```
head -n1 /home/cis90/???t*/P*/Shake*/s????t[13]
```

Parse the command above and identify:

1. The command: **head**
2. The options: **-n1**
3. The number of arguments: 4
4. The actual arguments:

```
/home/cis90/juetay/Poems/Shakespeare/sonnet1  
/home/cis90/juetay/Poems/Shakespeare/sonnet3  
/home/cis90/prites/Poems/Shakespeare/sonnet1  
/home/cis90/prites/Poems/Shakespeare/sonnet3
```

## Output from the command

```
/home/cis90/simben $ head -n1 /home/cis90/???t*/P*/Shake*/s????t[13]
==> /home/cis90/juetay/Poems/Shakespeare/sonnet1 <==
From fairest creatures we desire increase,

==> /home/cis90/juetay/Poems/Shakespeare/sonnet3 <==
Look in thy glass and tell the face thou viewest,

==> /home/cis90/prites/Poems/Shakespeare/sonnet1 <==
From fairest creatures we desire increase,

==> /home/cis90/prites/Poems/Shakespeare/sonnet3 <==
Look in thy glass and tell the face thou viewest,
/home/cis90/simben $
```

*Note: the shell expanded /home/cis90/???t\*/P\*/Shake\*/s????t[13] into four arguments which match the sonnet 1 and sonnet 3 files belonging to all students in the Spring 2015 class whose first name starts with a T*





# Meta Characters (review)

# Metacharacters

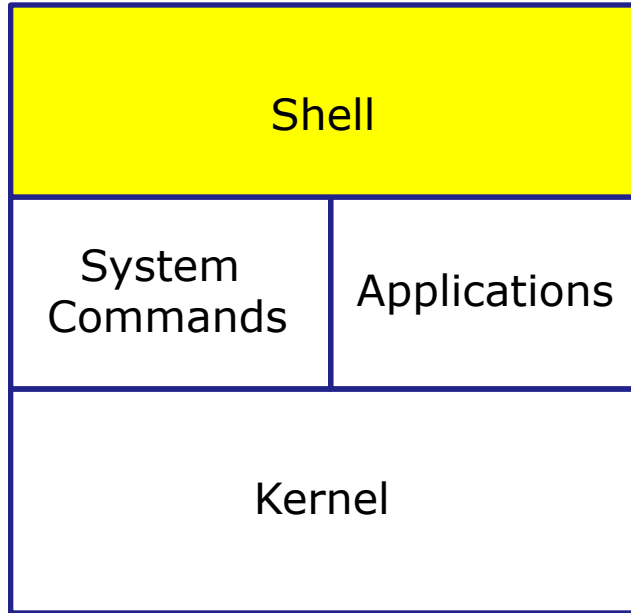
Have special interpretation by the shell

Char	Description
\	Treat the following metacharacter as a plain character. Also called "escaping" the next character.
\$	The following text is a shell (environment) variable and the value should be used.
<cr>	Carriage return marks the end of the command
;	Separates multiple commands on one line
'	used to enclose a string that the shell will not do further interpretation
"	Used to enclose a string that the shell will do further interpretation.
>	Redirects stdout ( <i>more in Lesson 8</i> )
2>	Redirects stderr ( <i>more in Lesson 8</i> )
*	Matches all non-hidden file names when used alone or zero or more characters when used as prefix, infix or postfix
?	Matches any single character of a file name
[]	Matches any single character contained within the brackets
#	Not an official metacharacter, but any text following the # is ignored by the shell



# Life of the Shell

The shell processes metacharacters during the **Parse** step



- 1) **Prompt** for a command
- 2) **Parse** (interpret metacharacters, expand file names and dissect command line into options, arguments and redirection)
- 3) **Search** for program (along the path)
- 4) **Execute** program by loading into memory (becomes a process), hookup input and outputs, and pass along command line options and arguments.
- 5) **Nap** (wait till process is done)
- 6) **Repeat**

# Metacharacters

#

*# has the ability to make everything that follows the # be ignored by the shell. Good for adding comments in scripts*

```
/home/cis90/simmsben $ #OK lets escape the carriage return in next example  
/home/cis90/simmsben $
```

*Note there is no error message because everything after the # is ignored*

# Metacharacters

\$

*\$ metacharacter has the ability to "show the value of"*

```
/home/cis90/simmsben $ EYES=brown  
/home/cis90/simmsben $ echo EYES  
EYES  
/home/cis90/simmsben $ echo $EYES  
brown  
  
/home/cis90/simmsben $ echo $LOGNAME  
simmsben  
/home/cis90/simmsben $
```

*echo the string EYES*

*echo the value of the variable EYES*

*echo the value of the predefined environment variable LOGNAME*

# Metacharacters " and '

*Weak "double" quotes allow the shell to process \$ metacharacters inside the quoted string*

```
/home/cis90/simmsben $ echo "I am in $PWD"  
I am in /home/cis90/simmsben
```

```
/home/cis90/simmsben $ echo 'I am in $PWD'  
I am in $PWD  
/home/cis90/simmsben $
```

*Strong "single" quotes block the shell from processing \$ metacharacters inside the quoted string*

# Metacharacters

;

```
/home/cis90/simmsben $ #Lets put two commands on one line  
/home/cis90/simmsben $ echo "This is my terminal device:"; tty  
This is my terminal device:  
/dev/pts/2  
/home/cis90/simmsben $
```

*the ; metachacter lets you combine several commands on one line*

# Metacharacters

\

```
/home/cis90/simmsben $ #OK lets escape the carriage return in next example  
/home/cis90/simmsben $ echo Lets start line 1 here \  
> and finish it here  
Lets start line 1 here and finish it here  
/home/cis90/simmsben $
```

*The \ is used to escape the next character typed. Use an escape to disable the special abilities of a metacharacter.*

*Escaping a carriage return (the Enter key) tells the shell to keep inputting more characters from the next line for the current command being entered.*



# Metacharacters

\

*Escaping the # means it is no longer treated as comment*

```

/home/cis90/simmsben $ \#OK lets put a comment here
-bash: #OK: command not found
/home/cis90/simmsben $
/home/cis90/simmsben $
/home/cis90/simmsben $ echo $PS1
$PWD $
/home/cis90/simmsben $ echo \ $PS1
$PS1
/home/cis90/simmsben $
    
```

*and you get an error when the shell processes your comment*

*Escaping the \$ means \$ is no longer treated "the value of"*



# Environment Variables

(review)

# Shell (Environment) Variables

## common environment variables

Shell Variable	Description
HOME	Users home directory (starts here after logging in and returns with a <code>cd</code> command (with no arguments))
LOGNAME	User's username for logging in with.
PATH	List of directories, separated by ':'s, for the Shell to search for commands (which are program files) .
PS1	The prompt string.
PWD	Current working directory
SHELL	Name of the Shell program being used.
TERM	Type of terminal device , e.g. dumb, vt100, xterm, ansi, etc.

# Shell (Environment) Variables

Show variable values

```
/home/cis90/simben $ echo $HOME  
/home/cis90/simben
```

```
/home/cis90/simben $ echo $LOGNAME  
simben90
```

```
/home/cis90/simben $ echo $PS1  
$PWD $
```

*Use echo to show the  
values of variables*

```
/home/cis90/simben $ echo $PWD  
/home/cis90/simben
```

```
/home/cis90/simben $ echo $SHELL  
/bin/bash
```

```
/home/cis90/simben $ echo $TERM  
xterm
```

# Shell (Environment) Variables

## PATH

```
/home/cis90/simben $ echo $PATH  
/usr/lib/qt-  
3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbi  
n:/sbin:/home/cis90/simben/../../bin:/home/cis90/simben/bin:.
```

*These are the directories in Benji's PATH in the order they will be searched:*

1<sup>st</sup>: /usr/lib/qt-3.3/bin  
2<sup>nd</sup>: /usr/local/bin  
3<sup>rd</sup>: /bin  
4<sup>th</sup>: /usr/bin  
5<sup>th</sup>: /usr/local/sbin  
6<sup>th</sup>: /usr/sbin  
7<sup>th</sup>: /sbin  
8<sup>th</sup>: /home/cis90/simben/../../bin  
9<sup>th</sup>: /home/cis90/simben/bin  
10<sup>th</sup>: .

*The PATH variable is used by the shell to locate commands*

# Shell (Environment) Variables

## Set variable values

*Use an "=" with no spaces to set values of variables*

```
/home/cis90/simben $ # Change the prompt variable
/home/cis90/simben $ PS1='[\u@\h \W]\$ '
[simben90@opus ~]$ echo $PS1
[\u@\h \W]\$
[simben90@opus ~]$
```

```
[simben90@opus ~]$ # Change it back again
[simben90@opus ~]$ PS1='$PWD $ '
/home/cis90/simben $ echo $PS1
$PWD $
/home/cis90/simben $
```

# Shell Variables

## Set variable values

*If the variable has never been used before then it is created*

```
/home/cis90/simben $ myfavoritedog="Benji"  
/home/cis90/simben $ echo $myfavoritedog  
Benji
```

# Shell (Environment) Variables

## env command – show all environment variables

```

/home/cis90/simmsben/Poems $ env
HOSTNAME=opus.cabrillo.edu
SHELL=/bin/bash
TERM=xterm
HISTSIZE=1000
USER=simmsben
LS_COLORS=no=00:fi=00:di=00;34:ln=00;36:pi=40;33:so=00;35:bd=40;33;01:cd=40;33;01:or=01;05;37;41:mi
=01;05;37;41:ex=00;32:* .cmd=00;32:* .exe=00;32:* .com=00;32:* .btm=00;32:* .bat=00;32:* .sh=00;32:* .csh=
00;32:* .tar=00;31:* .tgz=00;31:* .arj=00;31:* .taz=00;31:* .lzh=00;31:* .zip=00;31:* .z=00;31:* .Z=00;31:*
.gz=00;31:* .bz2=00;31:* .bz=00;31:* .tz=00;31:* .rpm=00;31:* .cpio=00;31:* .jpg=00;35:* .gif=00;35:* .bmp=
00;35:* .xbm=00;35:* .xpm=00;35:* .png=00;35:* .tif=00;35:
USERNAME=
MAIL=/var/spool/mail/simmsben
PATH=/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/cis90/simmsben/../../bin:/home/cis90/simmsbe
n/bin:.
INPUTRC=/etc/inputrc
PWD=/home/cis90/simmsben/Poems
LANG=en_US.UTF-8
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
SHLVL=1
HOME=/home/cis90/simmsben
BASH_ENV=/home/cis90/simmsben/.bashrc
LOGNAME=simmsben
CVS_RSH=ssh
LESSOPEN=|/usr/bin/lesspipe.sh %s
G_BROKEN_FILENAMES=1
_=/bin/env
OLDPWD=/home/cis90/simmsben
/home/cis90/simmsben/Poems $

```

*Use the **env** command to show all environment variables (a subset of the shell variables)*



# Shell Variables

## set command – show all shell variables

```
/home/cis90/simmsben/Poems $ set
```

```
BASH=/bin/bash
BASH_ARGC=()
BASH_ARGV=()
BASH_ENV=/home/cis90/simmsben/.bashrc
BASH_LINENO=()
BASH_SOURCE=()
BASH_VERSINFO=([0]="3" [1]="2" [2]="25" [3]="1"
[4]="release" [5]="i686-redhat-linux-gnu")
BASH_VERSION='3.2.25(1)-release'
COLORS=/etc/DIR_COLORS.xterm
COLUMNS=80
CVS_RSH=ssh
DIRSTACK=()
EUID=1160
GROUPS=()
G_BROKEN_FILENAMES=1
HISTFILE=/home/cis90/simmsben/.bash_history
HISTFILESIZE=1000
HISTSIZ=1000
HOME=/home/cis90/simmsben
HOSTNAME=opus.cabrillo.edu
HOSTTYPE=i686
IFS=$' \t\n'
IGNOREEOF=10
INPUTRC=/etc/inputrc
LANG=en_US.UTF-8
LESSOPEN='|/usr/bin/lesspipe.sh %s'
LINES=24
LOGNAME=simmsben
```

```
LS_COLORS='no=00:fi=00:di=00;34:ln=00;36:pi=40;33:so=00;35
:bd=40;33;01:cd=40;33;01:or=01;05;37;41:mi=01;05;37;41:ex=
00;32:*.cmd=00;32:*.exe=00;32:*.com=00;32:*.btm=00;32:*.ba
t=00;32:*.sh=00;32:*.csh=00;32:*.tar=00;31:*.tgz=00;31:*.a
rj=00;31:*.taz=00;31:*.lzh=00;31:*.zip=00;31:*.z=00;31:*.Z
=00;31:*.gz=00;31:*.bz2=00;31:*.bz=00;31:*.tz=00;31:*.rpm=
00;31:*.cpio=00;31:*.jpg=00;35:*.gif=00;35:*.bmp=00;35:*.x
bm=00;35:*.xpm=00;35:*.png=00;35:*.tif=00;35:'
MACHTYPE=i686-redhat-linux-gnu
MAIL=/var/spool/mail/simmsben
MAILCHECK=60
OLDPWD=/home/cis90/simmsben
OPTERR=1
OPTIND=1
OSTYPE=linux-gnu
PATH=/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/
cis90/simmsben/./bin:/home/cis90/simmsben/bin:.
PIPESTATUS=( [0]="0" )
PPID=26514
PROMPT_COMMAND='echo -ne
"\033]0;${USER}@${HOSTNAME}%.*}:${PWD/#$HOME/~}"; echo -ne
"\007"'
PS1='$PWD $'
PS2='> '
PS4='+ '
PWD=/home/cis90/simmsben/Poems
SHELL=/bin/bash
SHELLOPTS=braceexpand:emacs:hashall:histexpand:ignoreeof:i
nteractive-comments:monitor
SHLVL=1
SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpass
TERM=xterm
UID=1160
USER=simmsben
USERNAME=
_=env
consoletype=pty
```

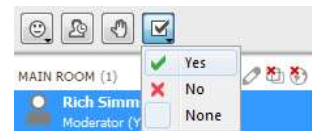
*Use the **set** command to show all shell variables (which includes the environment variables)*

## Class Exercise

- Change your prompt with:  
**PS1='\$LOGNAME, command please: '**
- Change your prompt with:  
**PS1='[\u@\h \W]\\$ '**
- Change your prompt with:  
**PS1="\$PWD \$ "**  
Now change directories using **cd**, what happened?
- Restore original prompt with:  
**PS1='\$PWD \$ '**

Did it work?

Click green "Yes" check or red "No" X  
on CCC Confer





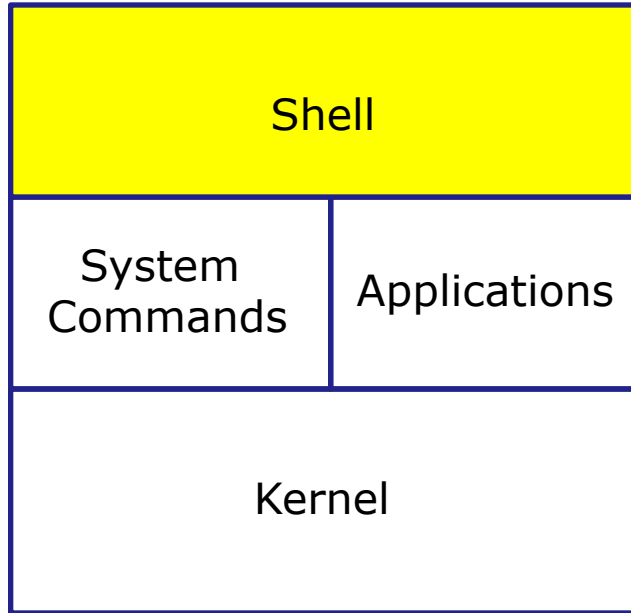
# Inputs & Outputs

## (continuing)



# Life of the Shell

*Execution begins only if the command is found*



- 1) **Prompt** for a command
- 2) **Parse** (interpret metacharacters, expand file names and dissect command line into options and arguments)
- 3) **Search** for program (along the path)
- 4) **Execute** program by loading into memory (becomes a process), hookup input and outputs, and pass along command line options and arguments.
- 5) **Nap** (wait till process is done)
- 6) **Repeat**



## Step 4 - the shell executes the command program file

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

*The next step is to load the **file** command that was found into memory. The program on the hard drive becomes a **process** in memory with a unique PID (Process ID).*

*Each new process is given three file descriptors **stdin**, **stdout** and **stderr** for input and output purposes.*

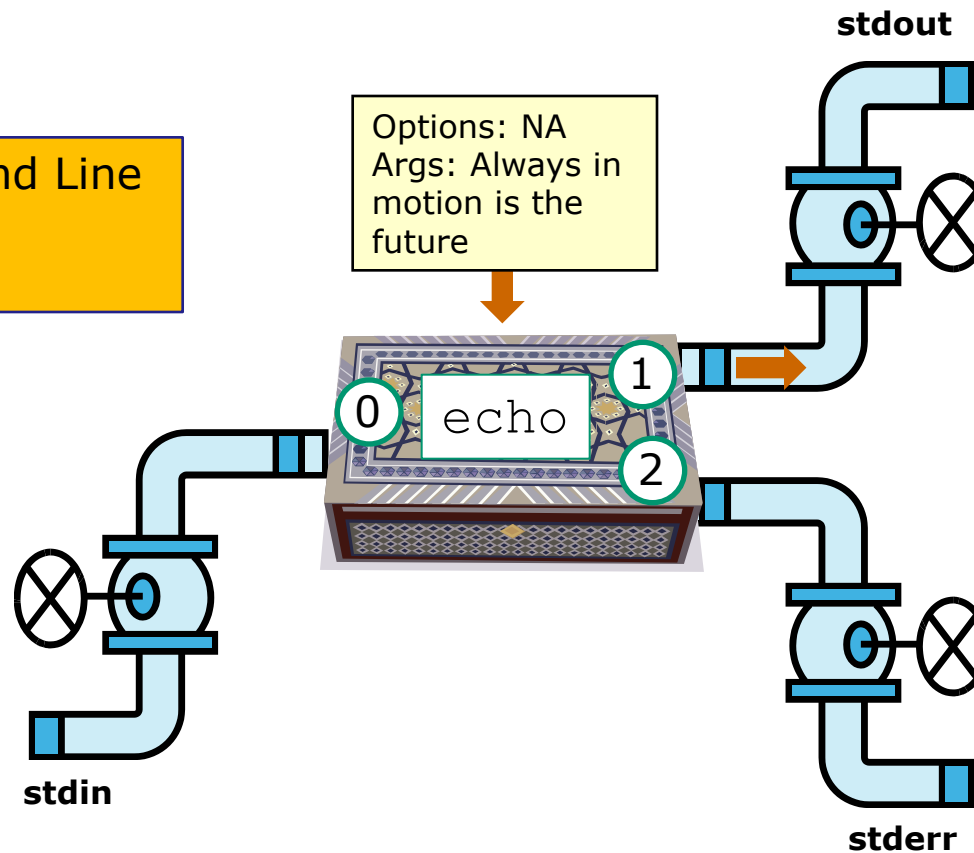
*These are sometimes referred to as the three **standard IO (Input/Output) streams**.*

## Example program to process: echo command

```
[rsimms@opus ~]$ echo Always in motion is the future
Always in motion is the future
[rsimms@opus ~]$
```

**Inputs:** Command Line

**Outputs:** stdout



`/dev/pts/1`



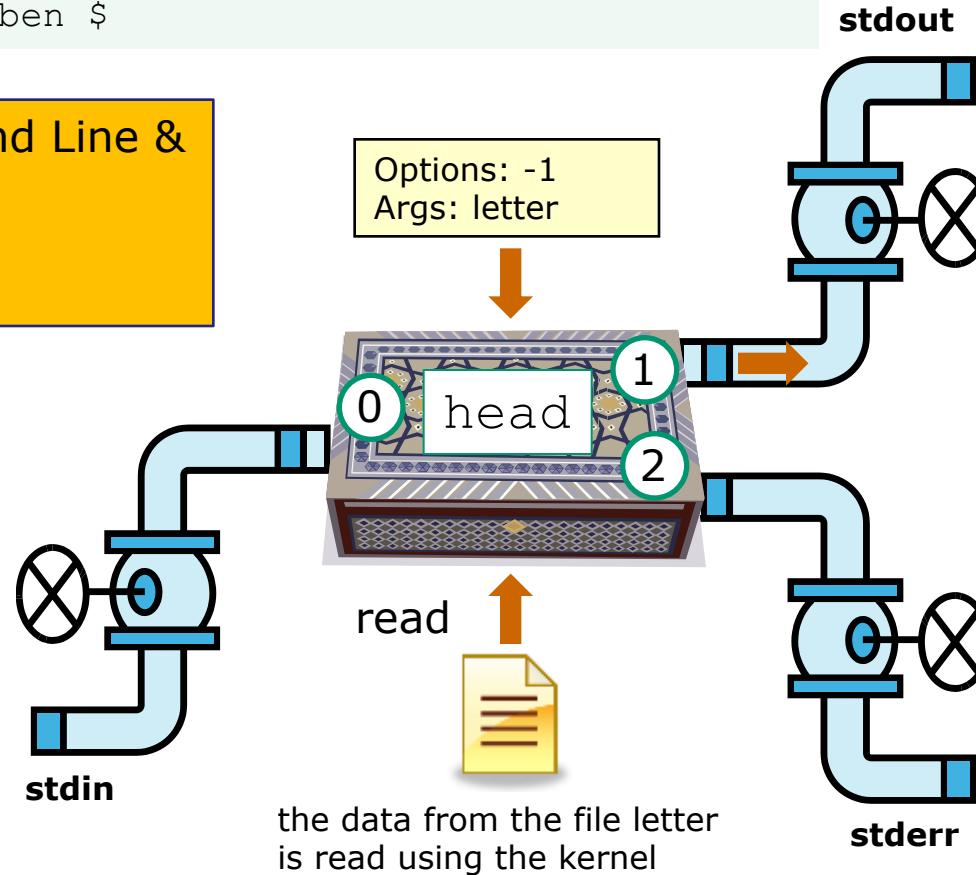
Always in  
motion is  
the future

## Example program to process: head command

```
/home/cis90/simmsben $ head -1 letter
Hello Mother! Hello Father!
/home/cis90/simmsben $
```

**Inputs:** Command Line & Operating System

**Outputs:** stdout



`/dev/pts/1`



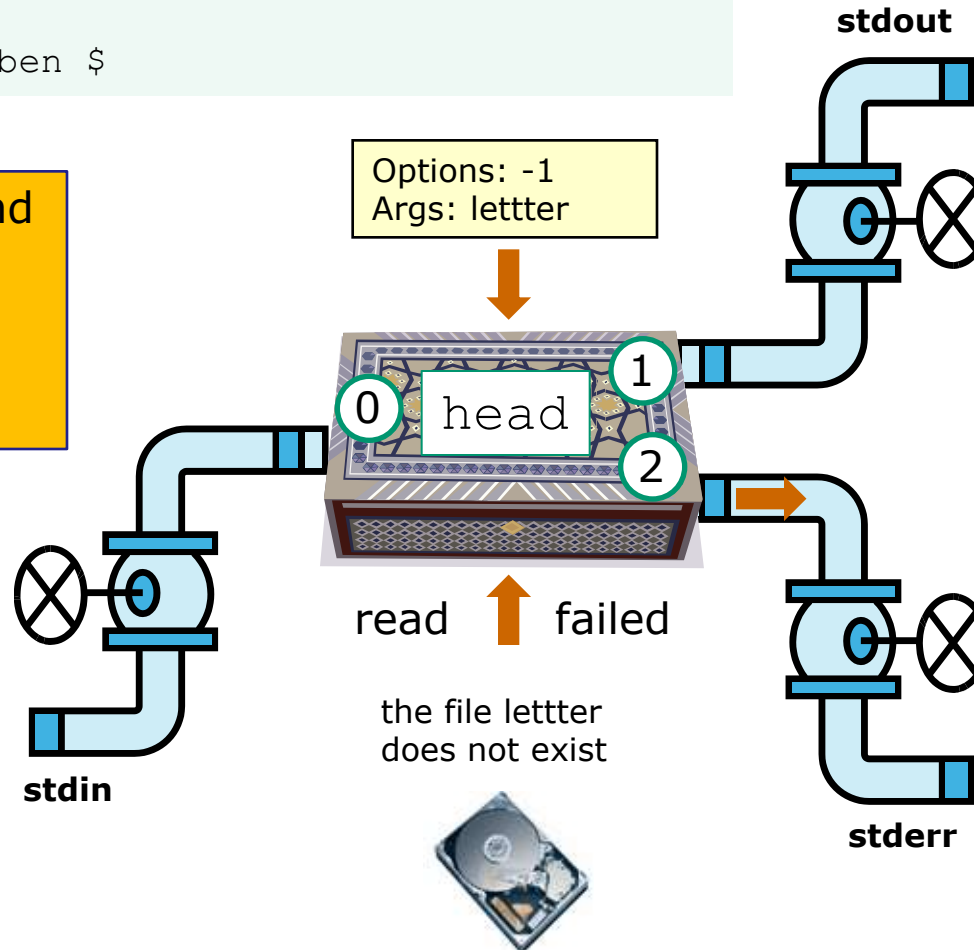
Hello Mother!  
Hello Father!

## Example program to process: head command

```
/home/cis90/simmsben $ head -1 lettter
head: cannot open `lettter' for reading: No such
file or directory
/home/cis90/simmsben $
```

**Inputs:** Command Line & Operating System

**Outputs:** stderr



/dev/pts/1



```
head: cannot
open `lettter'
for reading: No
such file or
directory
```



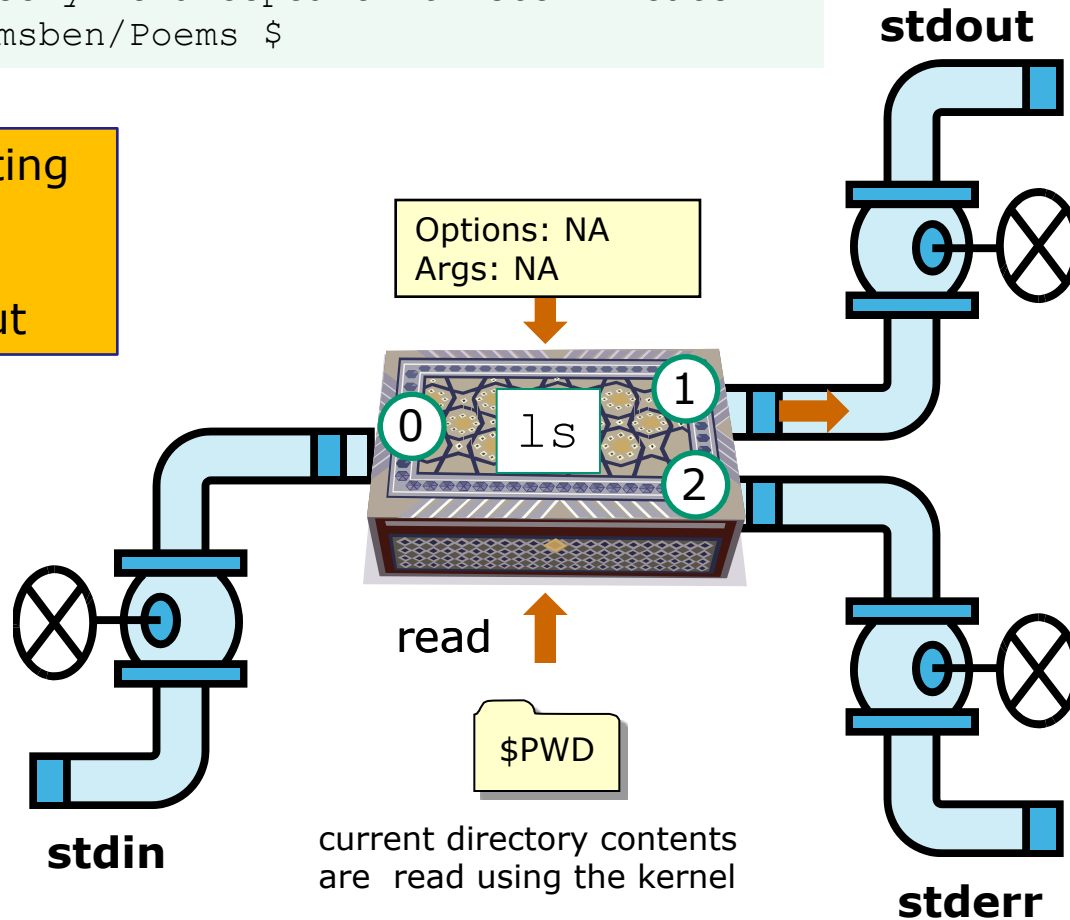
## Example program to process: ls command

```
/home/cis90/simmsben/Poems $ ls
ant Blake nursery Shakespeare twister Yeats
/home/cis90/simmsben/Poems $
```

`/dev/pts/1`

**Inputs:** Operating System

**Outputs:** stdout



```
ant Blake
nursery
Shakespeare
twister
Yeats
```

## Example program to process: bc command

```

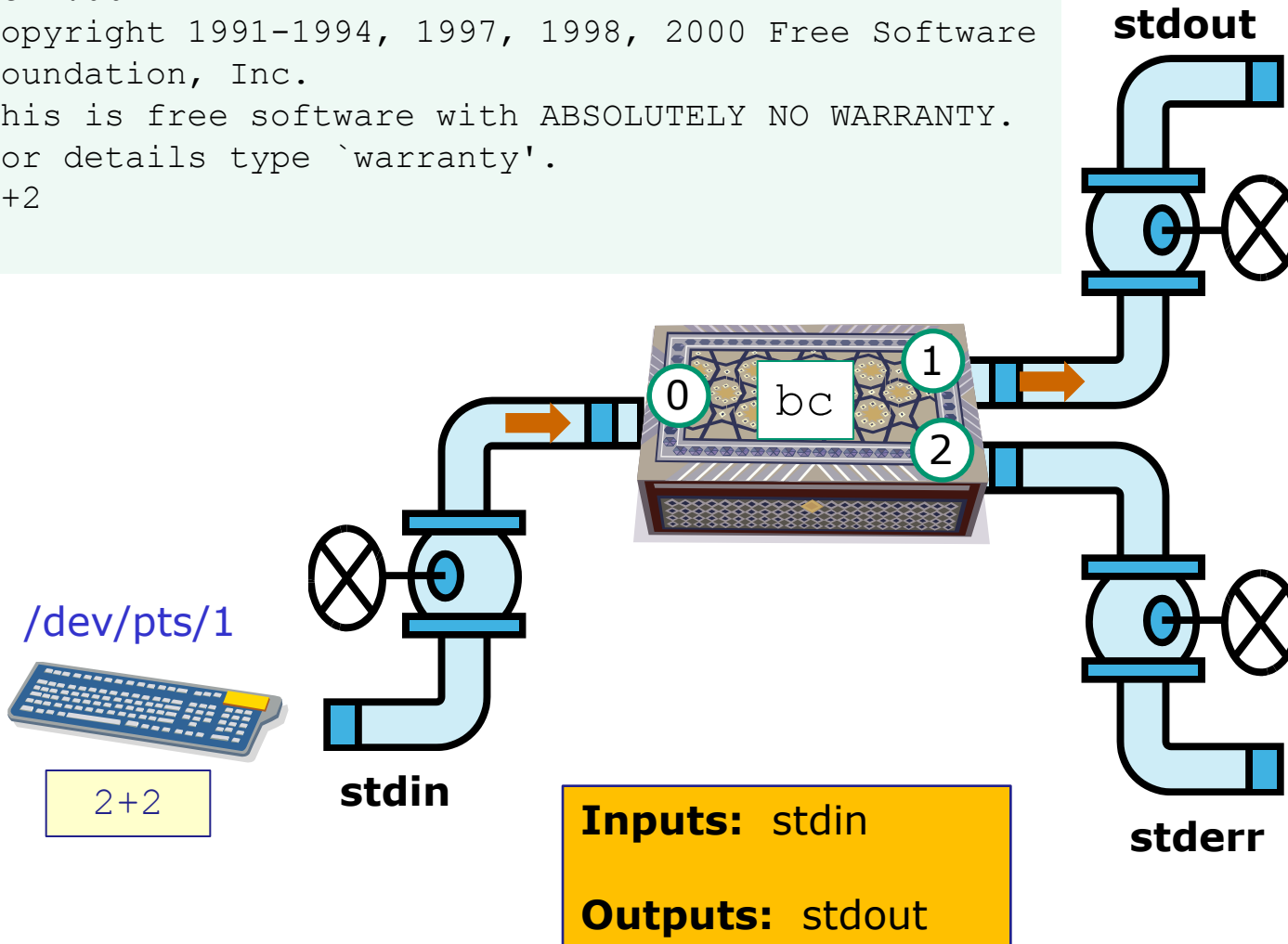
/home/cis90/simmsben $ bc
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free Software
Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
2+2
4
    
```

`/dev/pts/1`



```

bc 1.06
Copyright 1991-
1994, 1997,
1998, 2000 Free
Software
Foundation, Inc.
This is free
software with
ABSOLUTELY NO
WARRANTY.
For details type
`warranty'.
4
    
```

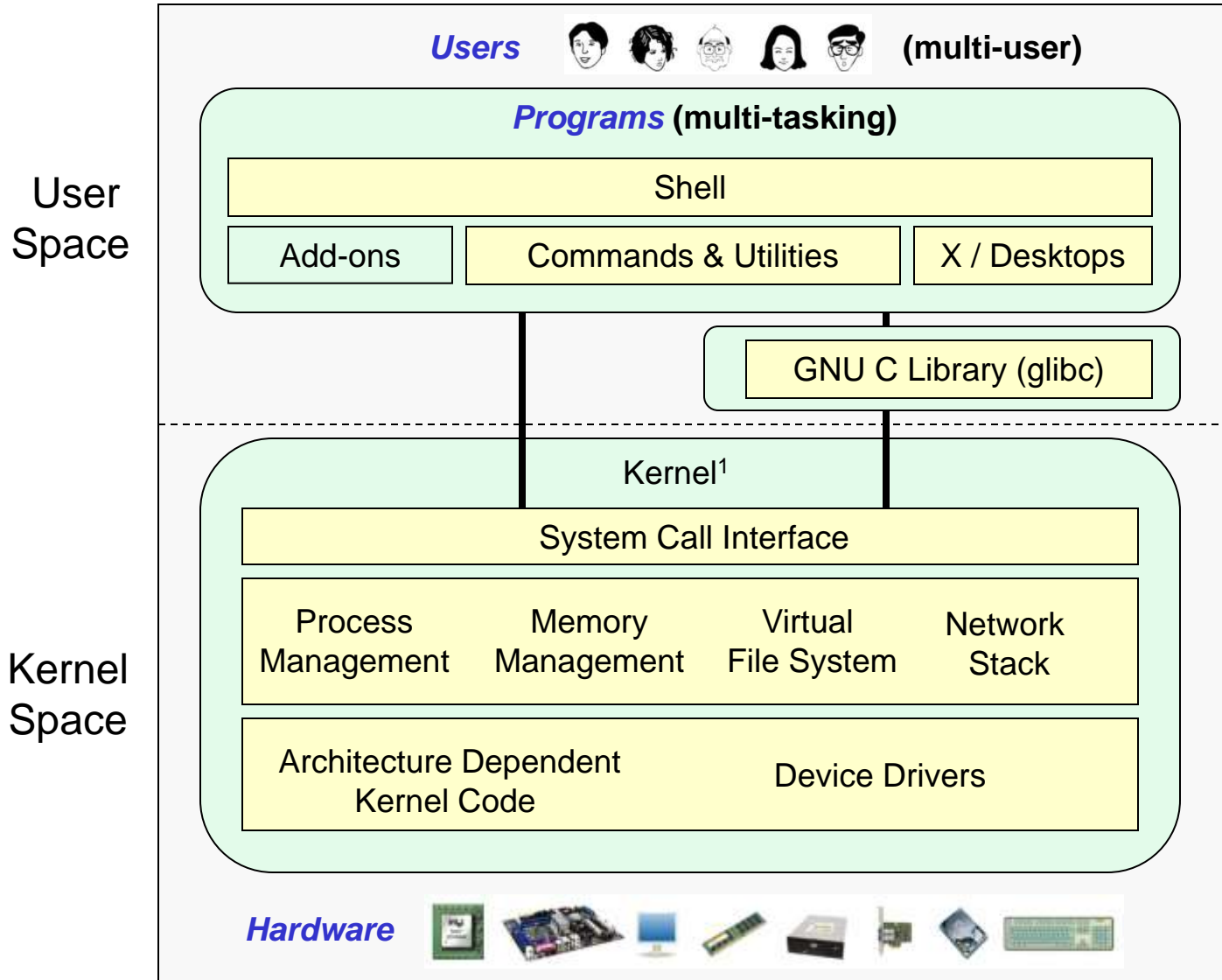




# Architecture (review)



GNU/Linux Operating System Architecture



Richard Stallman started the GNU project in 1983 to create a free UNIX-like OS. He Founded the Free Software Foundation in 1985. In 1989 he wrote the first version of the GNU General Public License



Linus Torvalds, as a student, initially conceived and assembled the Linux kernel in 1991. The kernel was later re-licensed under the GNU General Public License in 1992.

<sup>1</sup>See "Anatomy of the Linux kernel" by M. Tim Jones at <http://www-128.ibm.com/developerworks/linux/library/l-linux-kernel/>

# The Source for Linux Kernels



Use the **-r** option on **uname** to see which release of the kernel is running on your system

The first three numbers x.y.z indicate which kernel is being used from kernel.org.

x is the kernel version  
y is the major revision number  
z is the minor revision number

Anything after the dash was added by the distribution to indicate a modified kernel used by that distribution.

<https://www.kernel.org/>

```
[rsimms@oslab ~]$ uname -r
2.6.32-220.23.1.el6.i686
```

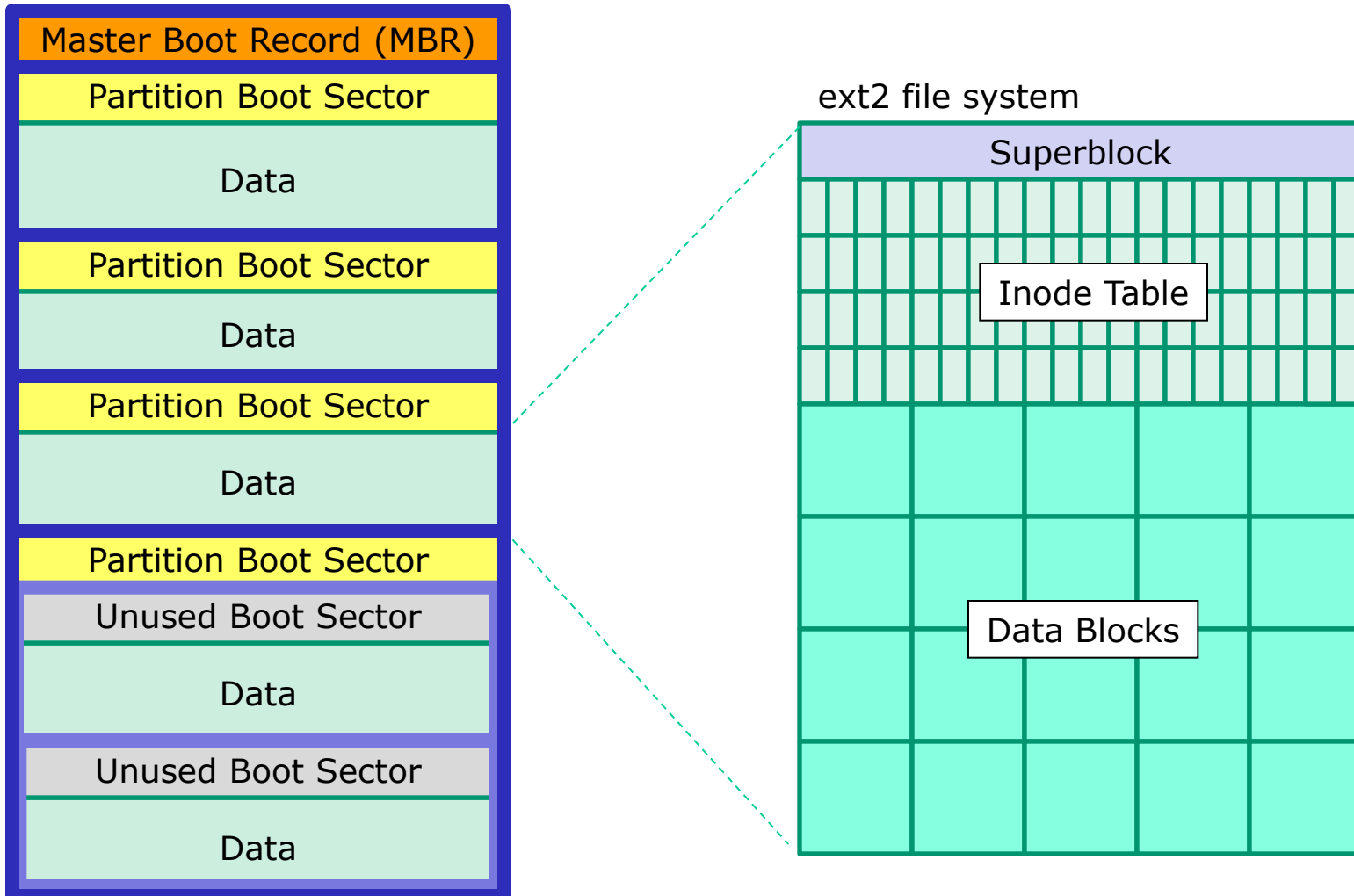
```
cis90@Arya-02:~$ uname -r
3.13.0-44-generic
```

*The 3.13.0 kernel on Arya is newer than the older 2.6.32 kernel on Opus*

# File System (review)

# File Systems

## Linux



# The three elements of a UNIX file

```
/home/cis90/simben/Poems $ ls
ant Blake nursery Shakespeare twister Yeats
```

```
/home/cis90/simben/Poems $ ls -li twister
102625 -rw-r--r-- 1 simben90 cis90 151 Jul 20 2001 twister
```

```
/home/cis90/simben/Poems $ cat twister
A tutor who tooted the flute,
tried to tutor two tooters to toot.
Said the two to the tutor,
"is it harder to toot? Or to
tutor two tooters to toot?"
```

filename

+

inode

+

data



filenames are stored in directories, **not** in inodes

bigfile 19470  
bin 9628  
letter 9662

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.

All the counselors hate the waiters, and the lake has alligators. You remember Leonard Skinner? He got ptomaine poisoning last night after dinner.

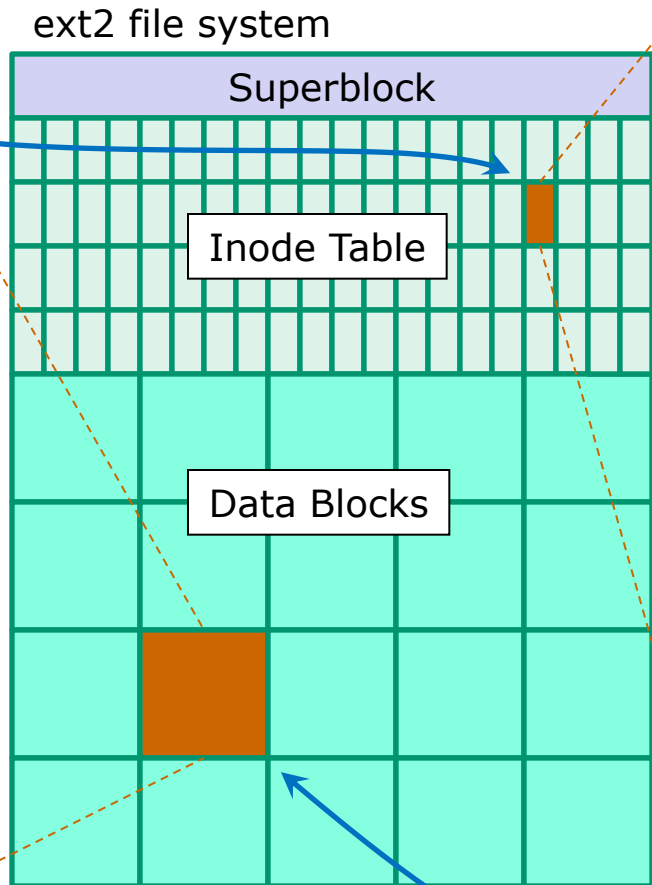
Now I don't want this to scare you, but my bunk mate has malaria. You remember Jeffrey Hardy? Their about to organize a searching party.

Take me home, oh Mother, Father, take me home! I hate Granada.  
Don't leave me out in the forest where I might get eaten by a bear! Take me home, I promise that I won't make noise, or mess the house with other boys, oh please don't make me stay -- I've been here one whole day.

Dearest Father, darling Mother, how's my precious little brother? I will come home if you miss me. I will even let Aunt Bertha hug and kiss me!

Wait a minute! It's stopped hailing! Guys are swimming!  
Guys are sailing! Playing baseball, gee that's better!  
Mother, Father, kindly disregard this letter.

Alan Sherman



9662	inode number
-	Type
rw-r--r--	Permissions
1	Number of links
simben90	User
cis90	Group
1044	Size
2001-07-20	Modification time
2012-09-17	Access Time
2012-08-01	Change time
Pointer(s) to data blocks	Pointer(s) to data blocks

```
/home/cis90/simmsben $ ls -il letter
9662 -rw-r--r--. 1 simben90 cis90 1044 Jul 20 2001 letter
```

# Basic File Types and Commands

Long listing code (ls -l)	Type	How to make one
d	directory	mkdir
-	regular <ul style="list-style-type: none"> <li>• Programs</li> <li>• Text</li> <li>• Data (binary)</li> </ul>	<i>Use the <b>file</b> command to further classify files</i> touch
l	symbolic link	ln -s
c	character device files	mknod
b	block device files	mknod

Note: Other files types includes sockets (s) and named pipes (p)

# Interpreting a long listing file types

```

simmsben@opus:~
/home/cis90/simmsben $ls -la
total 320
drwx----- 9 simmsben cis90 4096 Aug  8 11:51 .
drwxr-x--- 9 rsimms  cis90 4096 Jun 30 14:57 ..
-rw----- 1 simmsben cis90 11409 Aug  7 19:20 .bash_history
-rw----- 1 simmsben cis90   24 Jul 20 2001 .bash_logout
-rw----- 1 simmsben cis90  354 Sep 17 2003 .bash_profile
-rw----- 1 simmsben cis90  146 Jan 18 2004 .bashrc
-rw-rw-r-- 1 simmsben cis90   56 Jul  8 17:22 bcommands
-rw-r--r-- 2 simmsben cis90 10576 Jul 20 2001 bigfile
drwxr-xr-x 2 simmsben cis90 4096 Sep 11 2005 bin
-rw-rw-r-- 1 simmsben cis90 1044 Aug  8 11:52 deleteme
-rw-r--r-- 1 simmsben cis90  515 Jun 30 14:57 .emacs
-rw-r--r-- 1 simmsben cis90    0 Jul 20 2001 empty
d----- 2 simmsben cis90 4096 Feb  1 2002 Hidden
drwxr-xr-x 2 simmsben cis90 4096 Feb 17 2001 Lab2.0
drwxr-xr-x 3 simmsben cis90 4096 Feb 17 2001 Lab2.1
-rw----- 1 simmsben cis90   35 Aug  8 13:58 .lessht
-rw-r--r-- 1 simmsben cis90 1044 Jul 20 2001 letter
-rw----- 1 simmsben cis90 5799 Jul 24 21:08 mbox
drwxr-xr-x 2 simmsben cis90 4096 Sep 11 2005 Miscellaneous
-rw-r--r-- 1 simmsben cis90  759 Jun  6 2002 mission
drwxr-xr-x 4 simmsben cis90 4096 Jun 30 14:57 .mozilla
-rw-r--r-- 1 simmsben cis90   40 Jul 20 2001 .plan
drwxr-xr-x 5 simmsben cis90 4096 Jul  9 14:24 Poems
-rw-r--r-- 1 simmsben cis90 1074 Aug 26 2003 proposal1
-rw-r--r-- 1 simmsben cis90 2175 Jul 20 2001 proposal2
-rw-r--r-- 1 simmsben cis90 2054 Sep 14 2003 proposal3
-rw-r--r-- 1 simmsben cis90 5467 Jul  6 13:41 results-e1
-rw-r--r-- 1 simmsben cis90 1286 Jul  6 12:20 results-e1a
-rw-rw-r-- 1 simmsben cis90  688 Jul 24 15:35 salsa
-rw-r--r-- 1 simmsben cis90 1580 Nov 16 2004 small_town
-rw-r--r-- 1 simmsben cis90  485 Aug 26 2003 spellk
-rw-r--r-- 1 simmsben cis90  250 Jul 20 2001 text.err
-rw-r--r-- 1 simmsben cis90  231 Jul 20 2001 text.fxd
-rwxr-xr-x 1 simmsben cis90  509 Jun  6 2002 timecal
-rw----- 1 simmsben cis90  661 Jul 24 13:59 .viminfo
-rw-r--r-- 1 simmsben cis90  352 Jul 20 2001 what_am_i
-rw----- 1 simmsben cis90  126 Aug  7 14:23 .Xauthority
-rw-r--r-- 1 simmsben cis90   658 Jun 30 14:57 .zshrc
/home/cis90/simmsben $

```

All directories in the UNIX file tree contain these two hidden . and .. directories (d in column 1)

A regular file (- in column 1) Its hidden because it starts with a .

A directory (d in column 1) Color is blue because it's a directory

A hidden directory (d in column 1, name starts with .)

Regular file (- in column 1)

regular file (- in column 1) Color is green because with execute bits are set

## Symbolic links

*A symbolic link file  
(l in column 1)*

```
/home/cis90/simben $ ls -l accounts /etc/passwd
lrwxrwxrwx 1 simben90 cis90  11 Mar  7 08:52 accounts -> /etc/passwd
-rw-r--r-- 1 root      root  7183 Mar  6 08:17 /etc/passwd
/home/cis90/simben $
```

```
/home/cis90/simben $ head -5 /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

*The accounts file in Benji's directory is a symbolic link to the /etc/passwd file.*

```
/home/cis90/simben $ head -5 accounts
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

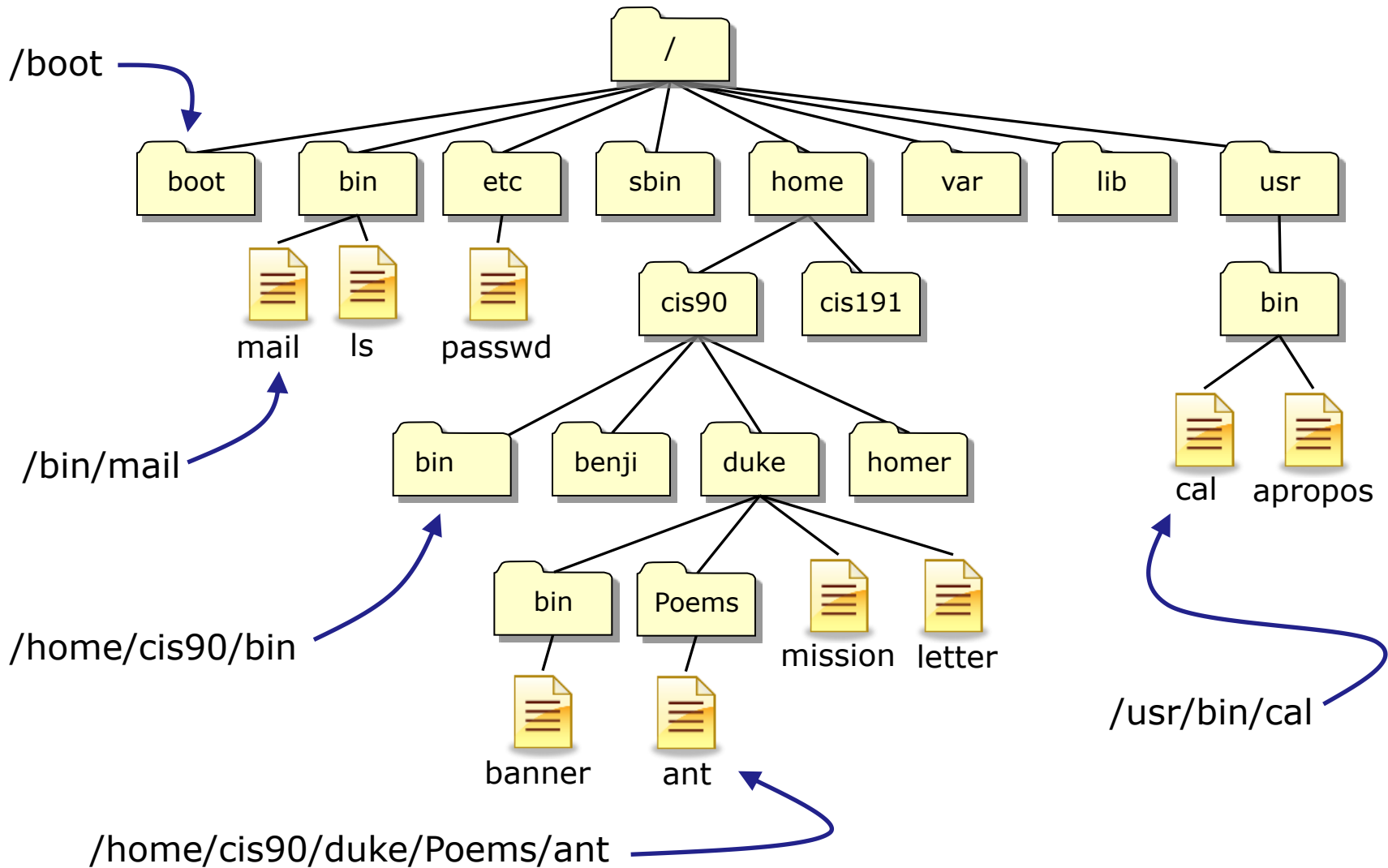
*These "shortcuts" can be used for convenience*

```
/home/cis90/simben $ ls -li accounts /etc/passwd
 99983 accounts 1280173 /etc/passwd
/home/cis90/simben $
```

*Note they have different inodes*

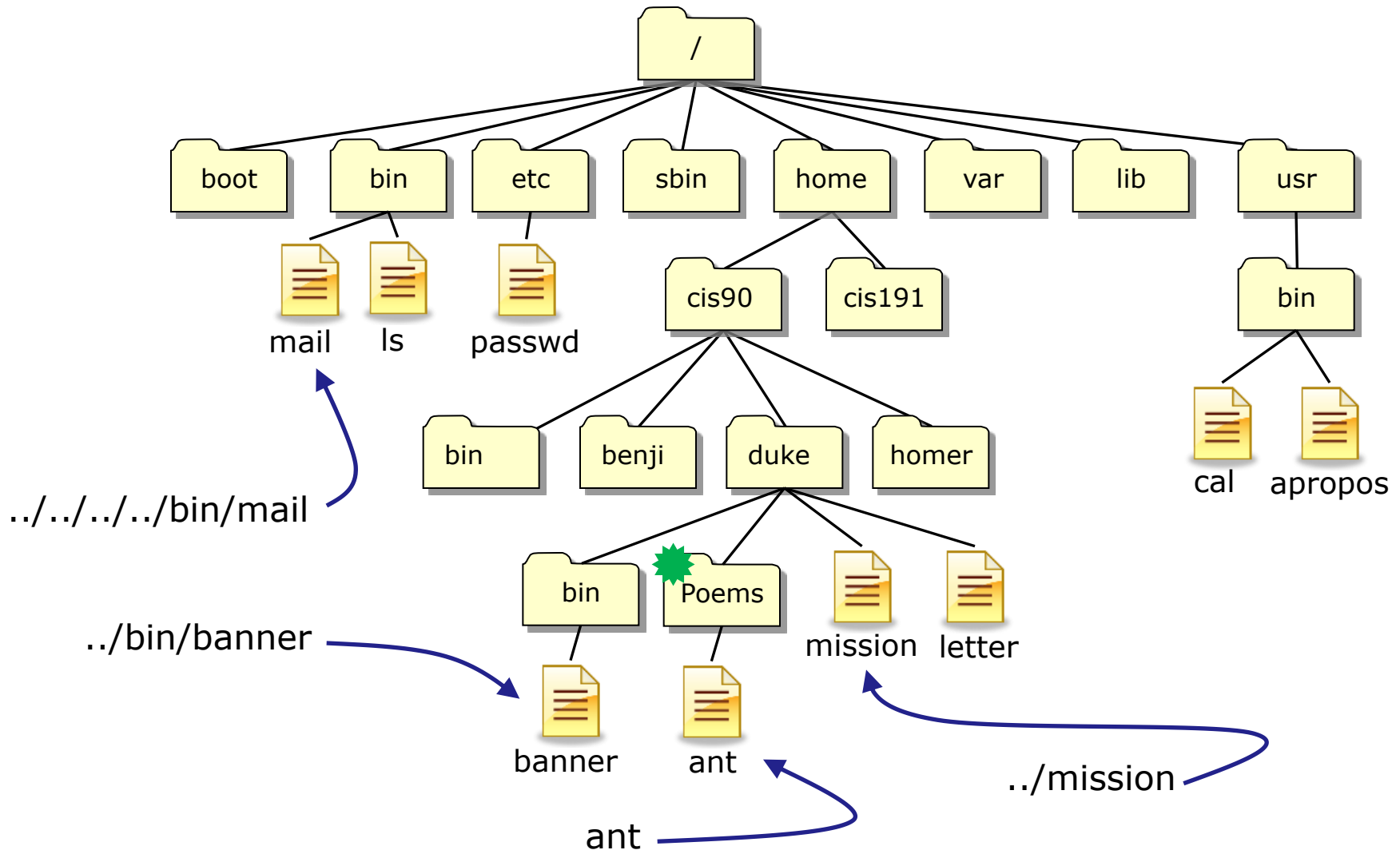
# Absolute Pathnames

Start with from /



# Relative Pathnames

Start from your current location in the tree



## Top Level Directories

Directory	Contents
/bin	binary files forming the commands and shells used by the system administrator and users
/boot	files used during the initial boot-up process including the kernel
/dev	device files for connected hardware
/etc	system configuration files
/home	individual directories owned by each user
/lib	shared libraries needed to boot the system and run the commands in the root filesystem (i.e. commands in /bin and /sbin)
/lost+found	recovered files that were corrupted by power failures or system crashes
/mnt	mount points for floppies, cds, or other file systems
/opt	add-on software packages and/or commercial applications
/proc	kernel level process information
/root	home directory for the root user
/sbin	system administration commands reserved for the superuser (root)
/tmp	temporary files that are deleted when the system is rebooted or started
/usr	program files and related files for use by all users
/var	log files, print spool files, and mail queues

## Absolute Pathname Target Practice



### **Analyze the absolute pathname**

What directory is the file in?

What is the name of the file in that directory?

*Type your answers in the chat window*





# CCC Confer

CCC Confer Breakout Rooms Test



Room 1



Room 2

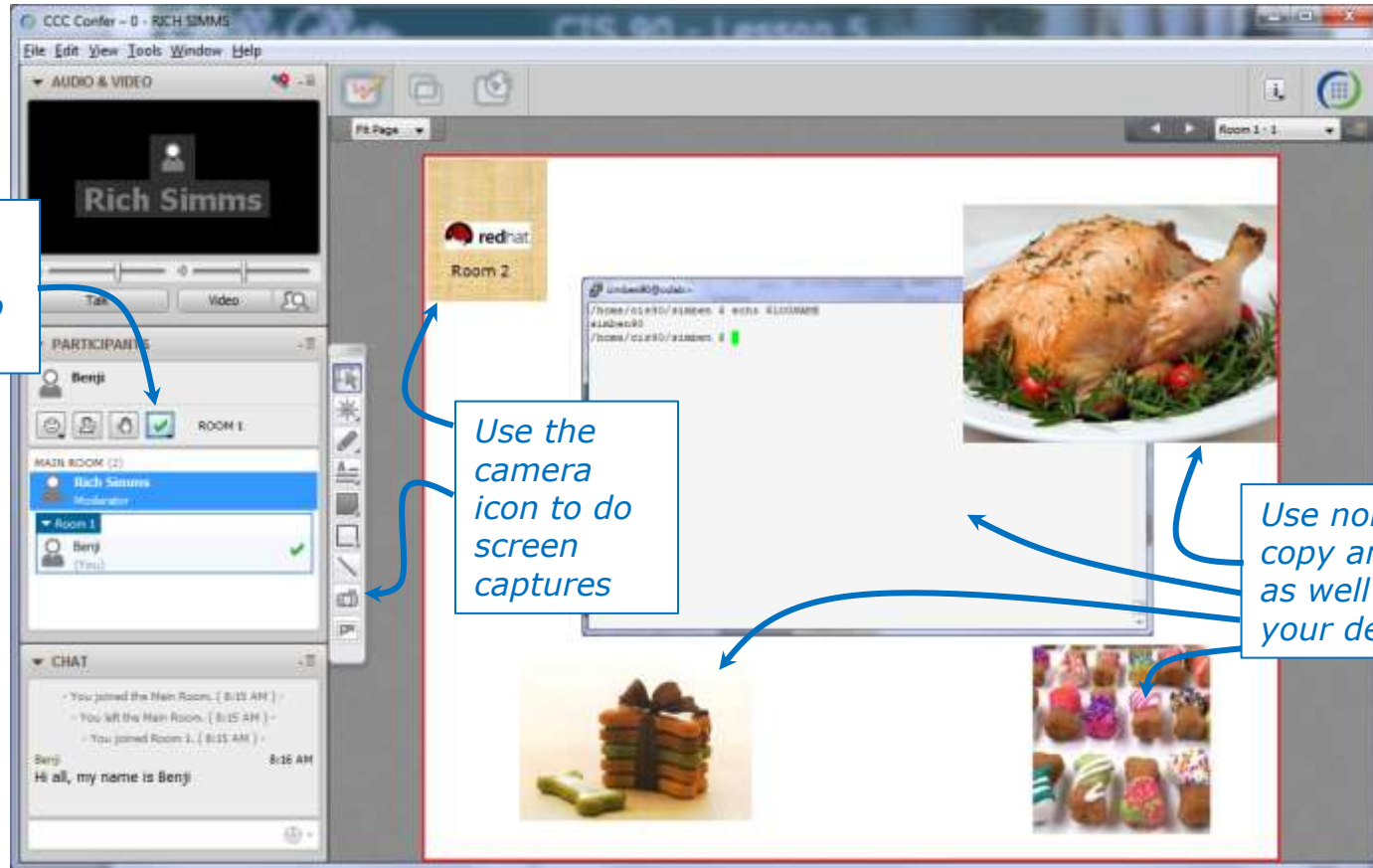


Room 3



Room 4

*Everyone needs to be on CCC Confer today,  
please use your Opus username.*



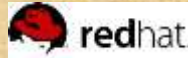
Use this to put a green check next to your name

Use the camera icon to do screen captures

Use normal copy and paste as well from your desktop

*I'll be sending you into virtual breakout rooms today so you can work together on various activities*

## CCC Confer Activity



Room 1



Room 2



Room 3



Room 4

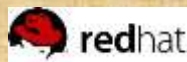
1. Download the presentation slides for Lesson 5 from the Calendar page of the web site.
2. Locate this slide.
3. Put a green check next to your name when you have done steps 1-2.

*When I see the green checks I'll distribute you the different rooms*

1. In your breakout room, see if you can do the following:
  - Introduce yourselves using room chat window.
  - Use whiteboard camera icon to copy your Linux logo above. Note you can resize the screen rectangle that is copied.
  - Each student use the **echo \$LOGNAME** command in a Putty/MAC terminal and then paste a copy of their ssh session on the whiteboard.
  - Decorate your room with anything else so you will recognize it when you return.
  - Return to the main room when finished (drag your name from the breakout room back to the main room)

# Flashcards

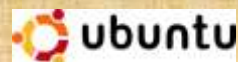
## Lessons L1-L5 random



Room 1



Room 2



Room 3



Room 4

Flashcards  
Deck size " "

L1-L5

L1=18

L2=22

L3=5

L4=26

L5=4

Total=75

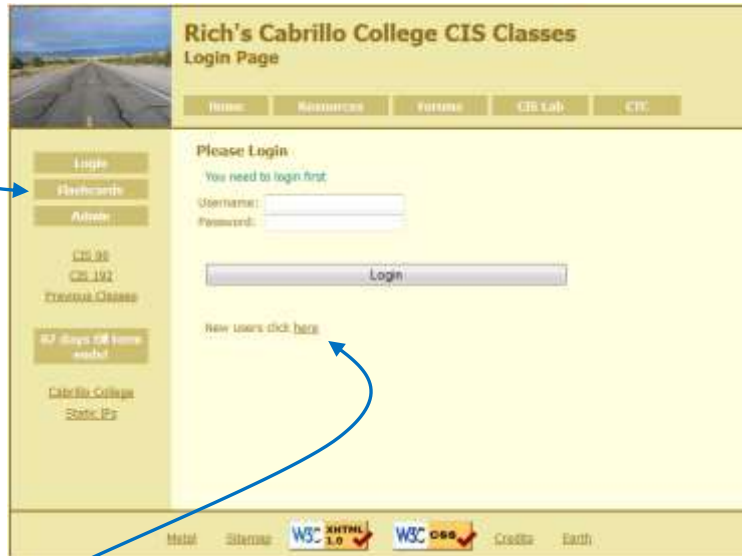
### Rules

- Chat window belongs to team that is up (no one else can use)
- "Final Answer" must be from someone on team that hasn't answered yet
- All team members can help each other and suggest answers

```
Instructor timer: /home/rsimms/scripts/countdown  
i=15; while [ $i -gt 0 ]; do clear; banner $i; let i=i-1; sleep 1; done; clear; banner done
```

# Flash Cards

*Click on Flashcards in left panel*



*Register if this is the first time using Flashcards*



*Register and choose a username and password of your choice*

# Logging in and using Flashcards

*Login with your username and password*

**Rich's Cabrillo College CIS Classes**  
Login Page

Home Resources Forums CIS Lab CTC

**Please Login**

Username:   
Password:

New users click [here](#)

87 days till term ends!

Cabrillo College: [Static IPs](#)

W3C XHTML 1.0 W3C CSS

*Select deck of cards*

**Rich's Cabrillo College CIS Classes**  
Select Flashcard Deck

Home Resources Forums CIS Lab CTC

**Select Card Deck**

"Random" decks are short, sweet and change everytime. The "All" decks include all the cards.

CIS 90	CIS 191
• Lesson 1 <a href="#">(Random)</a> <a href="#">(All)</a>	• Lesson 1 <a href="#">(Random)</a> <a href="#">(All)</a>
• Lesson 2 <a href="#">(Random)</a> <a href="#">(All)</a>	• Lesson 2 <a href="#">(Random)</a> <a href="#">(All)</a>
• Lesson 3 <a href="#">(Random)</a> <a href="#">(All)</a>	• Lesson 3 <a href="#">(Random)</a> <a href="#">(All)</a>
• Lesson 4 <a href="#">(Random)</a> <a href="#">(All)</a>	• Lesson 4 <a href="#">(Random)</a> <a href="#">(All)</a>
• Lesson 5 <a href="#">(Random)</a> <a href="#">(All)</a>	• Lesson 5 <a href="#">(Random)</a> <a href="#">(All)</a>
• Review 1-5 <a href="#">(Random)</a> <a href="#">(All)</a>	• Lesson 6 <a href="#">(Random)</a> <a href="#">(All)</a>
• Lesson 6 <a href="#">(Random)</a> <a href="#">(All)</a>	• Lesson 7 <a href="#">(Random)</a> <a href="#">(All)</a>
• Lesson 7 <a href="#">(Random)</a> <a href="#">(All)</a>	• Lesson 8 <a href="#">(Random)</a> <a href="#">(All)</a>
• Lesson 8 <a href="#">(Random)</a> <a href="#">(All)</a>	• Lesson 9 <a href="#">(Random)</a> <a href="#">(All)</a>
• Review 6-8 <a href="#">(Random)</a> <a href="#">(All)</a>	• Lesson 10 <a href="#">(Random)</a> <a href="#">(All)</a>
• Lesson 10 <a href="#">(Random)</a> <a href="#">(All)</a>	• Lesson 11 <a href="#">(Random)</a> <a href="#">(All)</a>
• Lesson 11 <a href="#">(Random)</a> <a href="#">(All)</a>	• Lesson 12 <a href="#">(Random)</a> <a href="#">(All)</a>
• Lesson 12 <a href="#">(Random)</a> <a href="#">(All)</a>	• Lesson 13 <a href="#">(Random)</a> <a href="#">(All)</a>
• Lesson 13 <a href="#">(Random)</a> <a href="#">(All)</a>	
• Lesson 14 <a href="#">(Random)</a> <a href="#">(All)</a>	
• Lesson 15 <a href="#">(Random)</a> <a href="#">(All)</a>	
• Review 10-15 <a href="#">(Random)</a> <a href="#">(All)</a>	
• All CIS 90 <a href="#">(Random)</a> <a href="#">(All)</a>	

87 days till term ends!

[Cabrillo College](#)  
[Static IPs](#)



## Class Exercise Flashcards

- Browse to [simms-teach.com](http://simms-teach.com)
- Register with a username and password of your choice
- Verify you can login and use the flash cards.



# Test Tips



# What command ... ?

## Tips on how to answer questions on lab assignments and tests

### What command will do “blah, blah, blah” questions:

Examples:

- What **ls** command allows you to see the permissions of your home directory while you are in your home directory?
- What command will give you a prompt showing your current working directory path and a \$?
- What command allows you to see hidden files in your current directory?

*Tip: Always use Opus (or the appropriate VM) to test your answers for these kinds of questions. **I will!** If your command doesn't work it won't be the right answer!*

## Practice Question

What **ls** command allows you to see the permissions of your home directory while you are in your home directory?

## Practice Question

What **ls** command allows you to see the permissions of your home directory while you are in your home directory?

```
/home/cis90/simben $ ls -l
total 392
-rw-r--r-- 2 simben90 cis90 10576 Jul 20 2001 bigfile
drwxr-xr-x 2 simben90 cis90 4096 Feb 12 16:07 bin
-rw----- 1 simben90 cis90 606 Feb 29 22:17 dead.letter
-rw-r--r-- 1 simben90 cis90 0 Jul 20 2001 empty
d----- 2 simben90 cis90 4096 Feb 1 2002 Hidden
< snipped >
-rw-r--r-- 1 simben90 cis90 250 Jul 20 2001 text.err
-rw-r--r-- 1 simben90 cis90 231 Jul 20 2001 text.fxd
-rwxr-xr-x 1 simben90 cis90 509 Jun 6 2002 timecal
-rw-rw-r-- 1 simben90 cis90 25390 Feb 29 22:18 uhistory

-rw-r--r-- 1 simben90 cis90 352 Mar 5 08:24 what_am_i
/home/cis90/simben $
```

**Nope, that didn't work.** We got permissions of all the files in the directory but we didn't get the permissions of the directory itself!

## Practice Question

What **ls** command allows you to see the permissions of your home directory while you are in your home directory?

```
/home/cis90/simben $ ls -dl /home/cis90/simben
drwxr-xr-x 10 simben90 cis90 4096 Mar  1 10:15
```

```
/home/cis90/simben $ ls -dl ~
drwxr-xr-x 10 simben90 cis90 4096 Mar  1 10:15
```

```
/home/cis90/simben $ ls -dl .
drwxr-xr-x 10 simben90 cis90 4096 Mar  1 10:15 .
```

```
/home/cis90/simben $ ls -dl $HOME
drwxr-xr-x 10 simben90 cis90 4096 Mar  1 10:15
```

```
/home/cis90/simben $ ls -dl
drwxr-xr-x 10 simben90 cis90 4096 Mar  1 10:15 .
```

***Yep, they all worked! The -d option instructs the ls command not to descend into the directory. Any of the commands above would be correct.***

## Practice Question

What command will give you a prompt showing your current working directory path and a \$?



## Practice Question

What command will give you a prompt showing your current working directory path and a \$?

```
/home/cis90/simben $ PS1=blah
blah
blahPS1="/home/cis90/simben $ "
```

```
/home/cis90/simben $
/home/cis90/simben $ cd ..
/home/cis90/simben $ cd
/home/cis90/simben $
/home/cis90/simben $ echo $PS1
/home/cis90/simben $
```

***Nope, that didn't work. The prompt doesn't change after changing to another directory***

## Practice Question

What command will give you a prompt showing your current working directory path and a \$?

```
/home/cis90/simben $ PS1=blah  
blah  
blahPS1="PWD $ "  
PWD $  
PWD $ echo $PS1  
PWD $
```

***Nope, that didn't work either. A \$ in front of the variable name is required to use its value.***

## Practice Question

What command will give you a prompt showing your current working directory path and a \$?

```
PWD $ PS1=blah
blah
blahPS1="$PWD $ "
/home/cis90/simben $ cd ..
/home/cis90/simben $ cd
/home/cis90/simben $
/home/cis90/simben $ echo $PS1
/home/cis90/simben $
```

***Better, but still didn't work.** The prompt is still not changing after cd'ing to another directory.*

*We need to block bash from expanding the \$PWD variable when it's being set.*

## Practice Question

What command will give you a prompt showing your current working directory path and a \$?

```
/home/cis90/simben $ PS1=blah
blah
blahPS1='$PWD $ '
/home/cis90/simben $ cd ..
/home/cis90 $ cd
/home/cis90/simben $
/home/cis90/simben $ echo $PS1
$PWD $
```

### **Touchdown! That worked!**

*The single quotes prevent bash from expanding \$PWD when setting the PS1 variable.*

*It is not expanded till the prompt is actually generated for the next command.*

## Practice Question

What command allows you to see hidden files in your current directory?

## Practice Question

What command allows you to see hidden files in your current directory?

```
/home/cis90/simben $ ls
bigfile          lab01.graded    Lab2.1          mission         small_town      uhistory
bin              lab01-submitted letter          Poems          spellk          what_am_i
dead.letter      lab02.graded    log             proposal1       text.err
empty            lab03.graded    mbox            proposal2       text.fxd
Hidden           Lab2.0          Miscellaneous    proposal3       timecal
```

***Nope, that didn't work! Hidden files start with a "." and note of these start with a "." (period)***

## Practice Question

What command allows you to see hidden files in your current directory?

```
/home/cis90/simben $ ls -a
.          dead.letter      Lab2.0        .mozilla      .ssh
..         .emacs            Lab2.1        .plan         text.err
.bash_history  empty           .lessht      Poems        text.fxd
.bash_logout  Hidden          letter        proposal1     timecal
.bash_profile lab01.graded    log           proposal2     uhistory
.bashrc       lab01-submitted mbox          proposal3     .viminfo
bigfile      lab02.graded    Miscellaneous  small_town   what_am_i
bin          lab03.graded    mission        spellk
```

**Bingo, that worked!** Hidden files and directories start with a "." (period)

How many arguments  
or “parse this  
command” questions



## Tips on how to answer questions on lab assignments and tests

### How many arguments or “parse this command” questions

Example: The shell performs file name expansion during the Parse step. When a user types the command: **file /v\*/l??/\*o\*.[14]** on Opus, how many arguments get passed to the **file** command? What specifically are those arguments?

*Tip: Use the echo command to preview how the shell will expand arguments containing metacharacters.*

***Tip:** Use the echo command to preview how the shell will expand arguments containing metacharacters.*

***TIP:** Use the echo command to preview how the shell will expand arguments containing metacharacters.*

***TIP:** Use the echo command to preview how the shell will expand arguments containing metacharacters.*

## Practice Question

The shell performs file name expansion during the Parse step. When a user types the command: **file /v\*/l??/\*o\*[14]** on Opus, how many arguments get passed to the **file** command? What specifically are those arguments?

```
/home/cis90/simben $ echo /v*/l??/*o*[14]  
/var/lib/polkit-1 /var/log/dracut.log-20130101 /var/log/yum.log-20130101
```

*Answer: The shell will expand **/v\*/l??/\*o\*[14]** into the 3 arguments shown above*

## Practice Question

Parse the following command on Opus:

```
wc -wl /home/cis90/d*t/*w*
```

what is the second argument passed to the **wc** command?

## Practice Question

Parse the following command on Opus:

```
wc -wl /home/cis90/d*t/*w*
```

what is the second argument passed to the **wc** command?

command: **wc**

options: **w** and **l**

arguments:

```
[rsimms@oslab ~]$ echo /home/cis90/d*t/*w*
```

```
/home/cis90/depot/network /home/cis90/depot/newfile /home/cis90/depot/randomwords
```

*3 arguments, the  
second argument is* 

*Answer: /home/cis90/depot/newfile*

## Practice Question

Parse the following command on Opus:

```
wc -wl /home/cis90/d*t/*w*
```

Regarding the options passed to the `wc` command, how many and what are they?

## Practice Question

Parse the following command on Opus:

```
wc -wl /home/cis90/d*t/*w*
```

Regarding the options passed to the wc command, how many and what are they?

command: wc

options: w and l

arguments:

/home/cis90/depot/network

/home/cis90/depot/newfile

/home/cis90/depot/randomwords

*Answer: there are two options, w and l*

# **Absolute / relative pathname questions:**

# Tips on how to answer questions on lab assignments and tests

## **Absolute/relative pathname questions:**

Examples:

- What is the relative pathname from your home directory to the **date** command?
- What is the absolute path to the sonnet1 file in your Shakespeare directory?

*Tip: Use the **ls** command with tab completion to verify your absolute or relative pathnames*

*Tip: Use the **ls** command with tab completion to verify your absolute or relative pathnames*

*Tip: Use the **ls** command with tab completion to verify your absolute or relative pathnames*

***Tip: Use the **ls** command with tab completion to verify your absolute or relative pathnames***



## Practice

What is the relative pathname from your home directory to the **date** command?

# Practice

What is the relative pathname from your home directory to the **date** command?

*First, use the type command to find where the date command is*

```
/home/cis90/simmsben $ type date
date is /bin/date
```

```
/home/cis90/simben $ ls ../
ahrmatt/      colabd/      huljef/      olscam/      rodduk/
answers/      deltas/      jimmel/      pacnan/      shidev/
.bash_profile depot/      lowmic/      phacha/      simben/
bin/          doucor/      macrya/      plajos/      varana/
blerav/       flammat/     maxsco/      plajua/      veleli/
bodian/       gueous/      mcidar/      porjon/
bunsol/       guest/       milhen/      pummas/
cheken/       helrog/      milhom/      rafdav/
cofcol/       hovdav/      milmic/      reedie/

/home/cis90/simben $ ls ../../
backup/      cis191/      cis90/      guest/      rick/      turnin/
cis164/      cis192/      cis98/      jimg/      rsimms/    .Xauthority
cis172/      cis193/      gerlinde/   mikki/      ryan/

/home/cis90/simben $ ls ../../..
.autofsck  etc/      media/      opt/      selinux/  tmp/
bin/       home/     misc/      proc/     srv/      u/
boot/      lib/     mnt/      root/     sys/      usr/
dev/       lost+found/ net/      sbin/     tftpboot/ var/

/home/cis90/simben $ ls ../../../../bin/date
../../../../bin/date
```

*Tap tab key twice to see what is in that directory*

*No errors so this relative pathname is GOOD!*

**Answer: ../../../../bin/date**

## Example

What is the absolute path to the sonnet1 file in your Shakespeare directory?

# Practice

What is the absolute path to the sonnet1 file in your Shakespeare directory?

```

/home/cis90/simben $ ls /
.autofsck  etc/      media/    opt/      selinux/  tmp/
bin/       home/    misc/     proc/     srv/      u/
< snipped >
/home/cis90/simben $ ls /home/
backup/    cis191/   cis90/    guest/    rick/     turnin/
< snipped >
/home/cis90/simben $ ls /home/cis90/
ahrmat/    colabd/   huljef/   olscam/   rodduk/
answers/   deltas/   jimmel/   pacnan/   shidev/
.bash_profile depot/    lowmic/   phacha/   simben/
< snipped >
cofcol/    hovdav/   milmic/   reedie/
/home/cis90/simben $ ls /home/cis90/simben/
.bash_history lab01.graded Miscellaneous/ .ssh/
< snipped >
.bashrc      lab03.graded .plan     timecal
bigfile      Lab2.0/       Poems/    uhistory
< snipped >
Hidden/     mbox       spellk
/home/cis90/simben $ ls /home/cis90/simben/Poems/
ant         Blake/     nursery   Shakespeare/ twister   Yeats/
/home/cis90/simben $ ls /home/cis90/simben/Poems/Shakespeare/sonnet
sonnet1  sonnet11  sonnet17  sonnet26  sonnet35  sonnet5  sonnet9
sonnet10 sonnet15  sonnet2   sonnet3   sonnet4   sonnet7
/home/cis90/simben $ ls /home/cis90/simben/Poems/Shakespeare/sonnet1
/home/cis90/simben/Poems/Shakespeare/sonnet1
  
```

*Tap tab key twice to see what is in that directory*

*No errors so this absolute pathname is GOOD!*

# Assignment





## How to prepare for the test:

- Review slides for Lessons 1-5 (download and make sure you know how to electronically search PDFs)
- DO THE PRACTICE TEST
- Compare your practice test answers with others and discuss on the forum
- DO THE PRACTICE TEST
- Note the steps you take to answer each question so you can use them again on the real test
- DO THE PRACTICE TEST
- Go through the Lesson 1-5 flashcards till you feel comfortable with the material
- DO THE PRACTICE TEST
- Practice, practice, practice ... repeating Labs 1-4 never hurts!

The practice test systems are no longer available after the real test starts



## Reminder to instructor:

On Sun-Hwa-IV

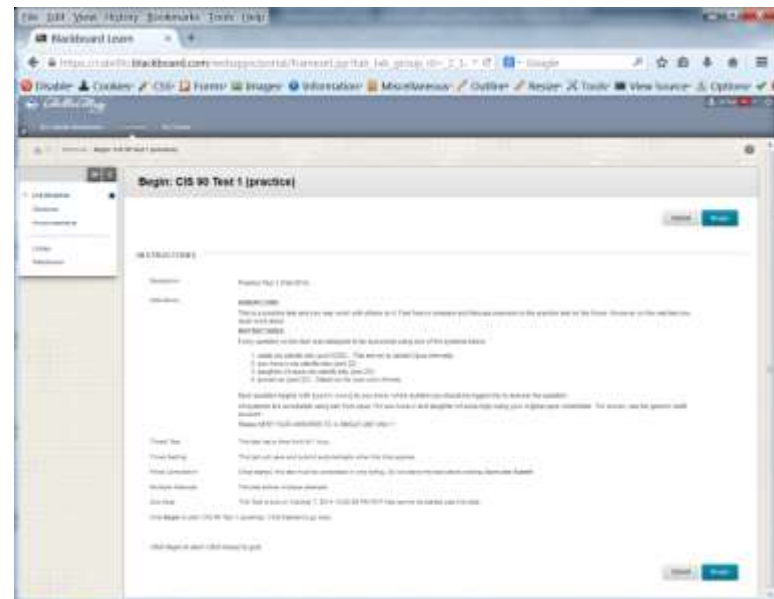
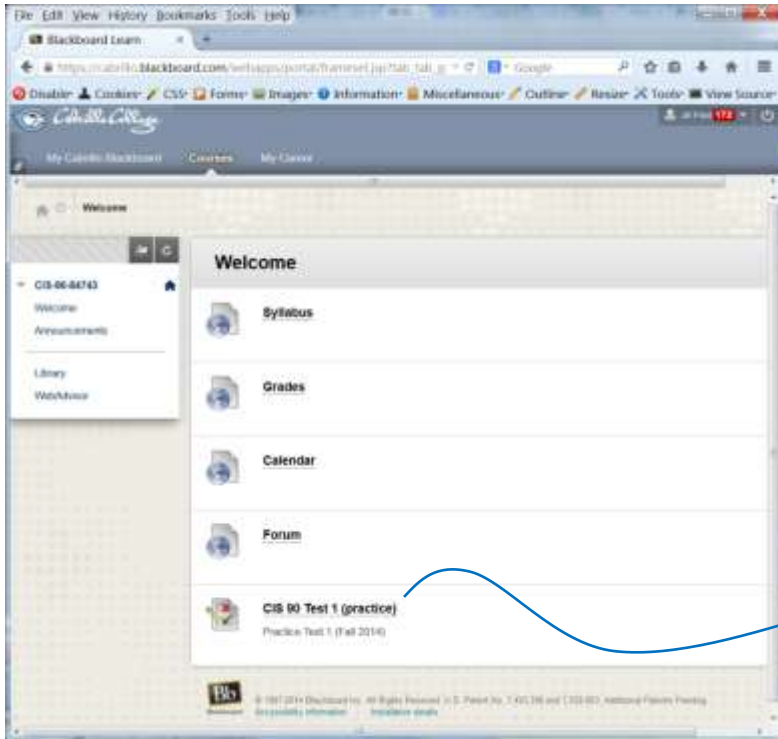
- create accounts
- run setup scripts (trouble-p1, setup-shakespeare)
- rm /etc/nologin

On Opus

- /home/rsimms/cis90/test01/q29/mail-q29-P1

# Practice Test

*A practice test is available on Blackboard.*







# Wrap up

New commands:

NA

NA

New metacharacters:

?

Matches any single character

[]

Matches any character in the brackets

New Files and Directories:

NA

NA

## Next Class

Assignment: Check Calendar Page on web site to see what is coming up.

No Quiz  
No Lab due  
Test!



# Backup

# ls command review

# ls command

Use the -l option for a "long listing"

1	2	3	4	5	6	7	8
simben90@opus:~							
/home/cis90/simben \$ ls -l							
total 308							
-rw-rw-r--	1	simben90	cis90	1870	Feb 24	15:37	1976
-rw-rw-r--	1	simben90	cis90	880	Feb 22	22:32	android
-rw-r--r--	2	simben90	cis90	10576	Jul 20	2001	bigfile
drwxr-xr-x	2	simben90	cis90	4096	Feb 12	16:07	bin
-rw-----	1	simben90	cis90	355	Feb 24	15:40	dead.letter
-rw-r--r--	1	simben90	cis90	0	Jul 20	2001	empty
d-----	2	simben90	cis90	4096	Feb 1	2002	Hidden
-r-----	1	simben90	staff	1182	Feb 16	13:17	lab01.graded
-rw-r--r--	1	simben90	cis90	494	Feb 12	16:39	lab01-submitted
-r-----	1	simben90	staff	1873	Feb 23	11:58	lab02.graded
drwxr-xr-x	2	simben90	cis90	4096	Feb 17	2001	Lab2.0
drwxr-xr-x	3	simben90	cis90	4096	Feb 17	2001	Lab2.1
-rw-r--r--	1	simben90	cis90	1044	Jul 20	2001	letter
-rw-r--r--	1	simben90	cis90	572	Feb 22	16:07	log
-rw-----	1	simben90	cis90	65469	Feb 26	14:44	mbox
drwxr-xr-x	2	simben90	cis90	4096	Sep 11	2005	Miscellaneous
-rw-r--r--	1	simben90	cis90	759	Jun 6	2002	mission
drwxr-xr-x	5	simben90	cis90	4096	Jan 18	2004	Poems
-rw-r--r--	1	simben90	cis90	1074	Aug 26	2003	proposal1
-rw-r--r--	1	simben90	cis90	2175	Jul 20	2001	proposal2
-rw-r--r--	1	simben90	cis90	2054	Sep 14	2003	proposal3
-rw-rw-r--	1	simben90	cis90	657	Feb 22	16:05	scott

total size of all files in blocks

*On Opus,  
1 block = 1024 bytes*

1. file type
  - = regular
  - d = directory
  - l = symbolic link
2. permissions
3. number of hard links
4. owner
5. group
6. size (in bytes)
7. last modified
8. filename

# ls command

## Using files vs directories as arguments

*Case 1: **No arguments** specified, all files in the current directory will be listed*

```
/home/cis90/simben $ ls
bigfile  Lab2.0          mission    proposal3  text.fxd
bin      Lab2.1           Poems     small_town timecal
empty    letter           proposal1 spellk      what_am_i
Hidden   Miscellaneous    proposal2 text.err
```

*Case 2: With a **filename** specified as an argument, just that file will be listed*

```
/home/cis90/simben $ ls bigfile
bigfile
```

*Case 3: With a **directory** specified as an argument, the contents of the directory will be listed*

```
/home/cis90/simben $ ls Poems/
ant  Blake  nursery  Shakespeare  twister  Yeats
```

# ls command

## specifying multiple directories

*The **ls** command can take multiple arguments*

*When a file is specified, just the filename is listed*

*When a directory is specified, the contents of the directory are listed*

```

/home/cis90/simben $ ls Poems/ bin/ letter
letter
bin/:
app banner enlightenment hi I treed tryme zoom

Poems/:
ant Blake nursery Shakespeare twister Yeats
    
```

*regular file* (arrow pointing to **letter**)

*directories* (arrows pointing to **Poems/** and **bin/**)



# ls command example

*The \* is expanded by the shell and replaced with the names of all files and directories in the current directory*

```

/home/cis90/simmsben $ ls *
bigfile  letter  proposal1  proposal3  spellk  text.fxd  what_am_i  Files listed
empty   mission proposal2  small_town  text.err  timecal   first
bin:
app  banner  enlightenment  hi  I  treed  tryme  zoom  Then the contents of
ls: Hidden: Permission denied  each directory are
                                   listed
Lab2.0:
386  A_long_name  file.9  READNAME  this_years_annual_report
afile  annual report  junk.old.bak  sTrAnGeNeSs

Lab2.1:
1.1  filename  junk  letter  more  old  Proposal3  Proposal.old  xyz

Miscellaneous:
better_town  file.dos  fruit  manpage  mystery  salad

Poems:
ant  Blake  nursery  Shakespeare  twister  Yeats

```

*Do you see the error message? ... permission issue (more in future lessons)*  
*Do you see the symbolic link? ... in light blue (more in future lessons)*

# ls command

## How to override showing directory contents

*The contents of the directory are shown*

```
/home/cis90/simben $ ls bin  
app  banner  enlightenment  hi  I  treed  tryme  zoom
```

*The directory itself is shown with the -d option*

```
/home/cis90/simben $ ls -d bin  
bin
```

*Use the **d** option to list the directory itself. Without the **d** the directory contents are listed instead.*

# ls command

## How to override showing directory contents

*The directory contents are shown*

```
/home/cis90/simben $ ls -i bin  
9634 app 9635 banner 9636 enlightenment 9630 hi 9632 I  
9631 treed 9633 tryme 9629 zoom
```

*The directory itself is shown with the -d option*

```
/home/cis90/simben $ ls -id bin  
9628 bin
```

*Use the **d** option to list the directory itself.*

# ls command

## Recursively list subdirectories (-R)

**ls -lR**

```

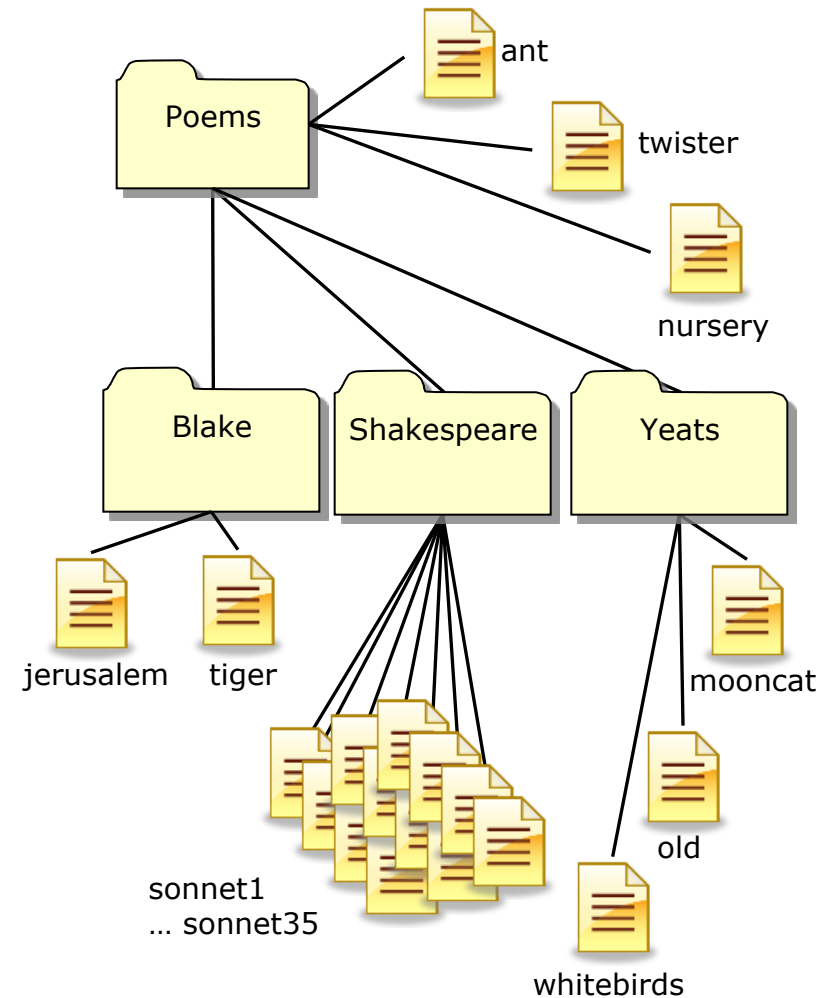
simmsben@opus:~/Poems
[simmsben@opus Poems]$ls -lR
.:
total 48
-rw-r--r-- 1 simmsben cis90 237 Aug 26 2003 ant
drwxr-xr-x 2 simmsben cis90 4096 Jul 20 2001 Blake
-rw-r--r-- 1 simmsben cis90 779 Oct 12 2003 nursery
drwxr-xr-x 2 simmsben cis90 4096 Oct 31 2004 Shakespeare
-rw-r--r-- 1 simmsben cis90 151 Jul 20 2001 twister
drwxr-xr-x 2 simmsben cis90 4096 Jul 20 2001 Yeats

./Blake:
total 16
-rw-r--r-- 1 simmsben cis90 582 Jul 20 2001 jerusalem
-rw-r--r-- 1 simmsben cis90 115 Jul 20 2001 tiger

./Shakespeare:
total 104
-rw-r--r-- 1 simmsben cis90 614 Jul 20 2001 sonnet1
-rw-r--r-- 1 simmsben cis90 620 Jul 20 2001 sonnet10
-rw-r--r-- 1 simmsben cis90 689 Oct 31 2004 sonnet11
-rw-r--r-- 1 simmsben cis90 618 Jul 20 2001 sonnet15
-rw-r--r-- 1 simmsben cis90 647 Jul 20 2001 sonnet17
-rw-r--r-- 1 simmsben cis90 631 Jul 20 2001 sonnet2
-rw-r--r-- 1 simmsben cis90 601 Jul 20 2001 sonnet26
-rw-r--r-- 1 simmsben cis90 615 Jul 20 2001 sonnet3
-rw-r--r-- 1 simmsben cis90 598 Jul 20 2001 sonnet35
-rw-r--r-- 1 simmsben cis90 588 Jul 20 2001 sonnet4
-rw-r--r-- 1 simmsben cis90 622 Jul 20 2001 sonnet5
-rw-r--r-- 1 simmsben cis90 581 Jul 20 2001 sonnet7
-rw-r--r-- 1 simmsben cis90 620 Jul 20 2001 sonnet9

./Yeats:
total 24
-rw-r--r-- 1 simmsben cis90 855 Jul 20 2001 mooncat
-rw-r--r-- 1 simmsben cis90 520 Jul 20 2001 old
-rw-r--r-- 1 simmsben cis90 863 Jul 20 2001 whitebirds
[simmsben@opus Poems]$

```



## Class Exercise

- Go to your home directory, type: **cd**
- Do a long listing of every file in your home directory and sub-directories and include inode numbers

**ls -l Miscellaneous/**

**ls -ld Miscellaneous/**

**ls -lR**