

Lesson Module Checklist

- Slides
- WB converted fro PowerPoint
- Flash cards
- Properties
- Page numbers
- 1st minute quiz
- Web Calendar summary
- Web book pages
- Commands
- Lab tested (check Q1 kernel release number and finger user account)
- Opus - submit windows locked at 00:00:00 deadline
- Bring Add Codes
- Bring printed roster
- Backup slides, whiteboard slides, handouts on flash drive
- 9V backup battery for microphone
- Key card for door



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 (How to attend from home or in the classroom)

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

Student checklist (How to layout your screen when attending class)

Google

CCC Confer

Downloaded PDF of Lesson Slides

The screenshot displays a virtual classroom environment with several overlapping windows:

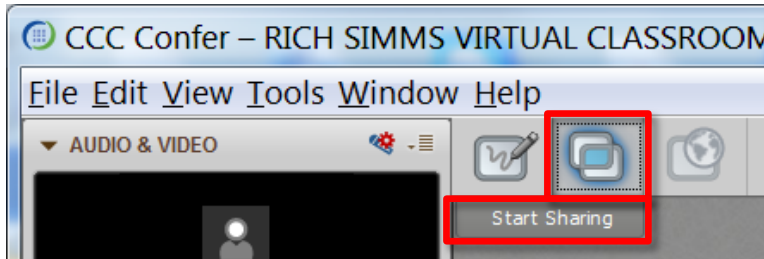
- Blackboard Co. x Google**: A browser window showing the **Rich's Cabrillo College CIS 90 Calendar** page. The page includes a sidebar with links like **Login**, **Flashcards**, **Admin**, and a main content area with a calendar table. A blue arrow points from the **Google** label to this window.
- CCC Confer - RICH SIMMS VIRTUAL CLASSROOM**: A video conferencing window showing a video feed of **Rich Simms** and a list of participants including **Benji Simms** and **Rich Simms**. A blue arrow points from the **CCC Confer** label to this window.
- cis90lesson01.pdf - Adobe Acrobat Pro**: A PDF window titled **CIS 90 - Lesson 1** showing a slide titled **The CIS 90 System Playground**. A blue arrow points from the **Downloaded PDF of Lesson Slides** label to this window.
- Google**: A map window showing a location in **Cabrillo College, San Jose, CA**. A blue arrow points from the **Google** label to this window.
- edu's password:**: A terminal window showing a login session with the text: `edu's password: 14:21 2015 from c-71-204-162-14` and `(^v^)`. A blue arrow points from the **One or more login sessions to Opus** label to this window.

CIS 90 website
Calendar page

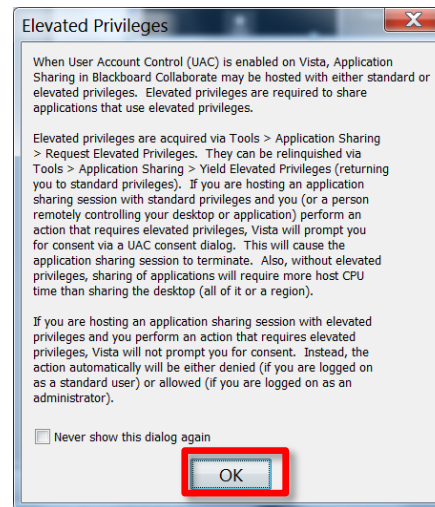
One or more login
sessions to Opus

Student checklist (To share your desktop with the class)

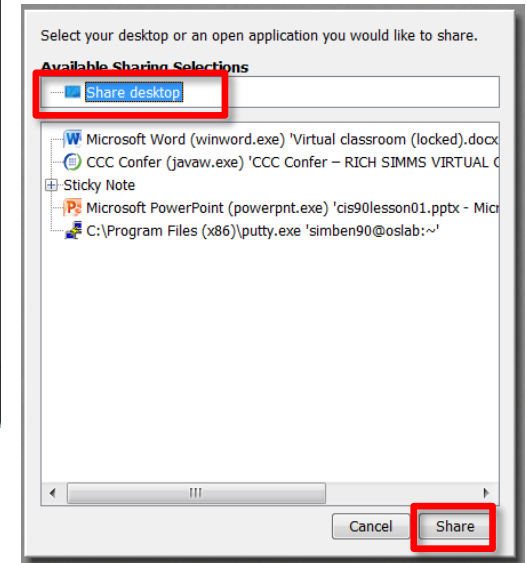
1) Instructor gives you sharing privileges



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.



Instructor: **Rich Simms**

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

Passcode: **136690**



Al



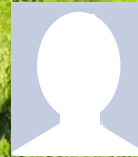
Tim



Emilio



Chris



Eddie



Clara



Ryan



Ethan



Monte



Cameron



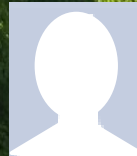
Tanner



Tess



Ahmad



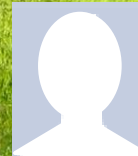
Mike F.



Mario



Rebecca



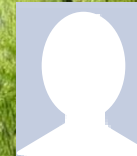
Diego



John



Efrain



Ivers



Michael S.



Django



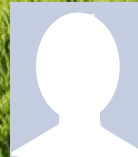
Ian



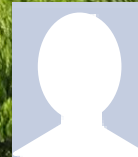
Abraham C.



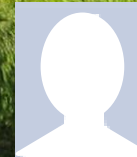
Benji



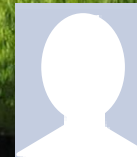
Abraham N.



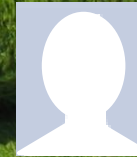
Roberto



Mateo



Melissa



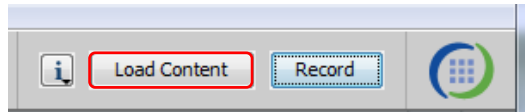
Sean



Rich's CCC Confer checklist - setup

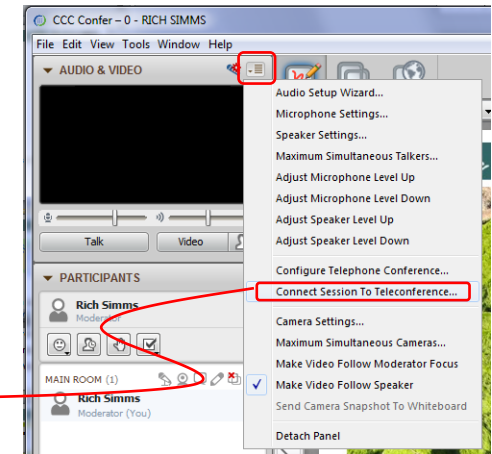
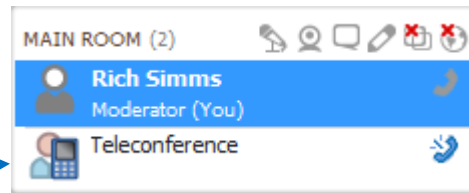


[] Preload White Board

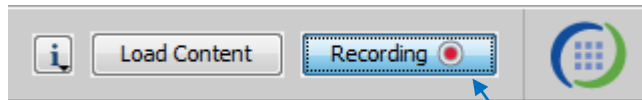


[] Connect session to Teleconference

Session now connected to teleconference



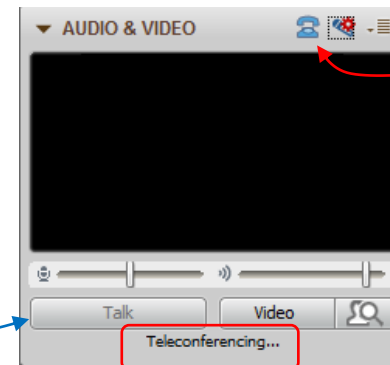
[] Is recording on?



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 - app layout



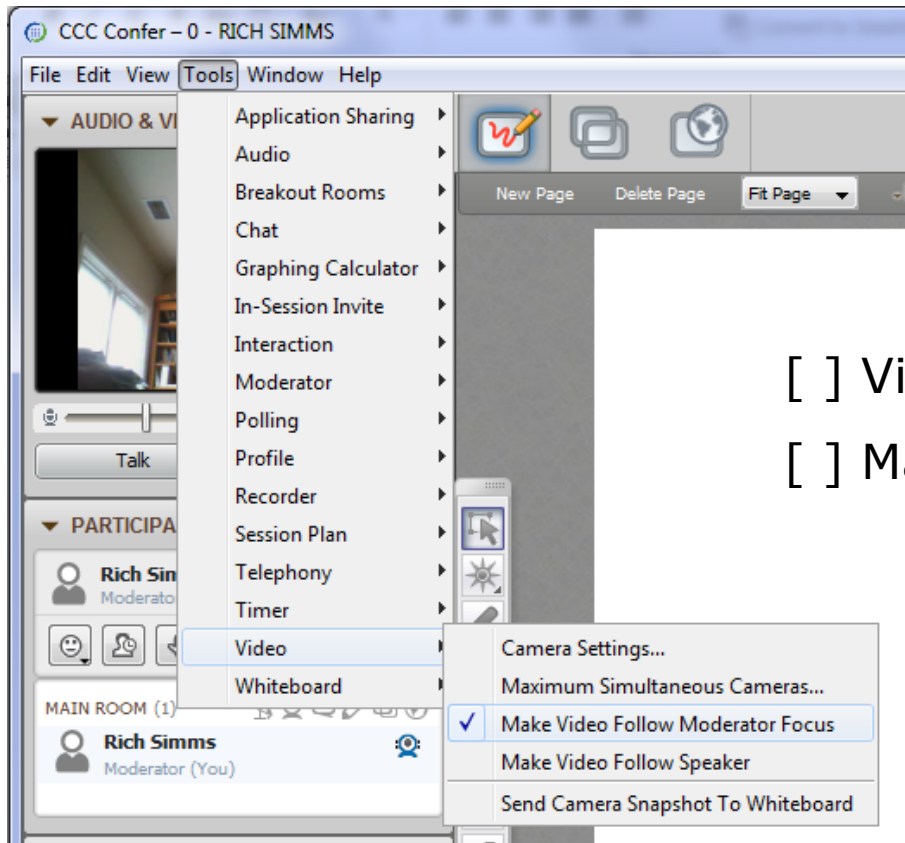
The screenshot displays a Windows desktop environment with several applications open. A red box labeled "foxit for slides" points to a Foxit Reader window displaying a PDF document. Another red box labeled "chrome" points to a Google Chrome window showing a web page with flashcard questions. A third red box labeled "vSphere Client" points to a vSphere Client window showing a list of virtual machines. A fourth red box labeled "putty" points to a PuTTY terminal window showing a login prompt. The desktop also features a CCC Confer window, a file explorer, and a taskbar with various icons.

[] layout and share apps





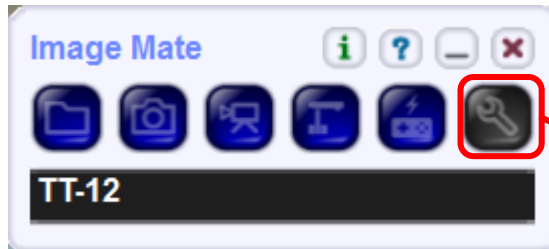
Rich's CCC Confer checklist - video



[] Video (webcam)

[] Make Video Follow Moderator Focus

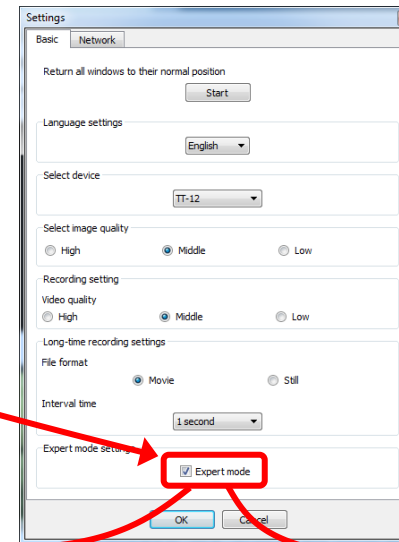
Rich's CCC Confer checklist - Elmo



Elmo rotated down to view side table



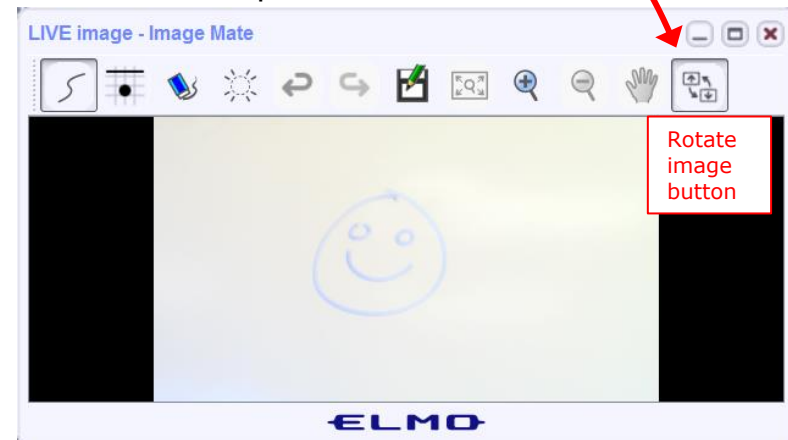
Run and share the Image Mate program just as you would any other app with CCC Confer



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



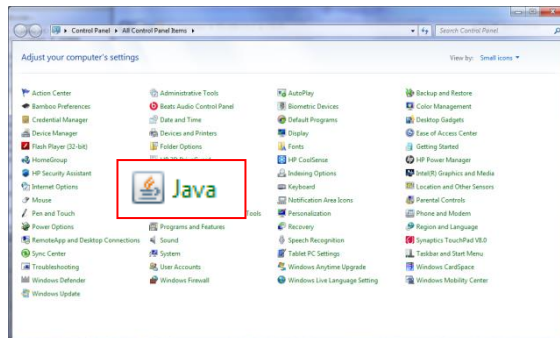
Rich's CCC Confer checklist - universal fix



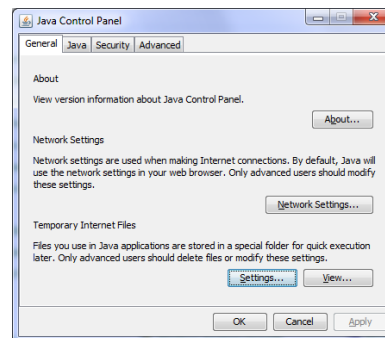
Universal Fix for CCC Confer:

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

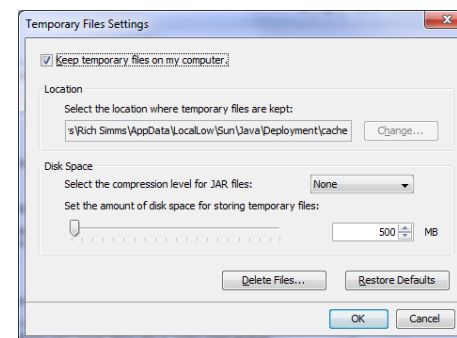
Control Panel (small icons)



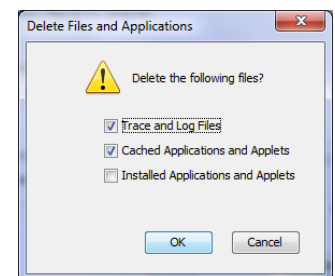
General Tab > Settings...



500MB cache size



Delete these



Google Java download



First Minute Quiz

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

Use CCC Confer White Board

email answers to: risimms@cabrillo.edu

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



Commands

Objectives	Agenda
• Understand how the UNIX login operation works. • Meet John the Ripper and learn how vulnerable a poor password is. • Understand basic command syntax and operation. • Understand program files and what happens when they are run. • Understand how the shell works and environment variables. • Understand how to get documentation when online.	• Quiz • Questions • Logging in • Passwords • Housekeeping • New commands & locations • Programs & inputs • Command line syntax & parsing • Variables • Shell six steps • Metacharacters • Path • Docs • Wrap up

Questions

Questions

How this course works?

Past lesson material?

Previous labs?

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.

FYI

CIS 90 and Smartphones (Android)



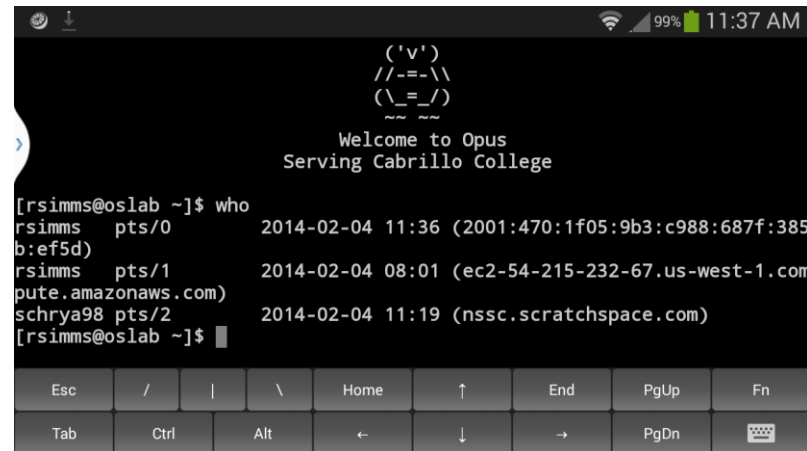
Blackboard
Collaborate App



*Join CCC Confer
virtual classroom*



JuiceSSH - SSH Client app

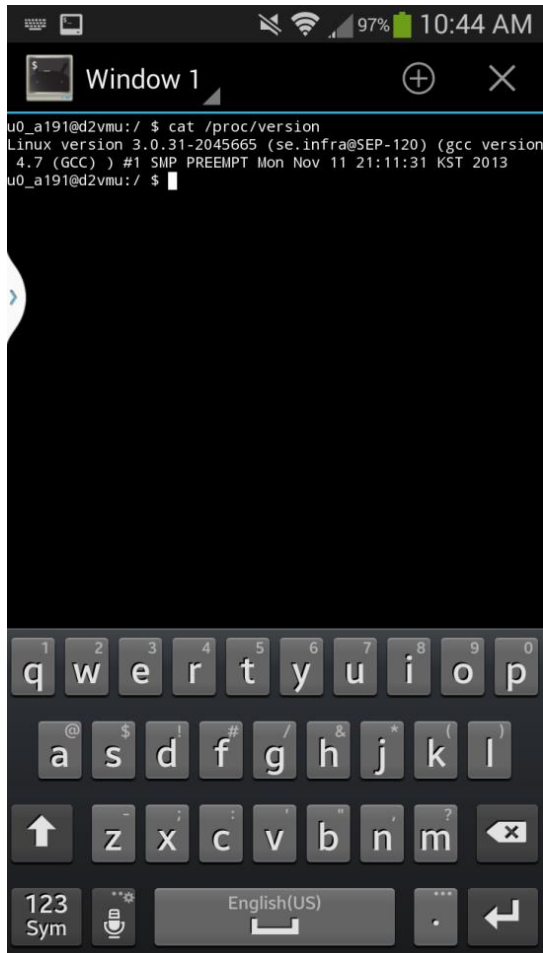


Login to to Opus

CIS 90 and Smartphones (Android)



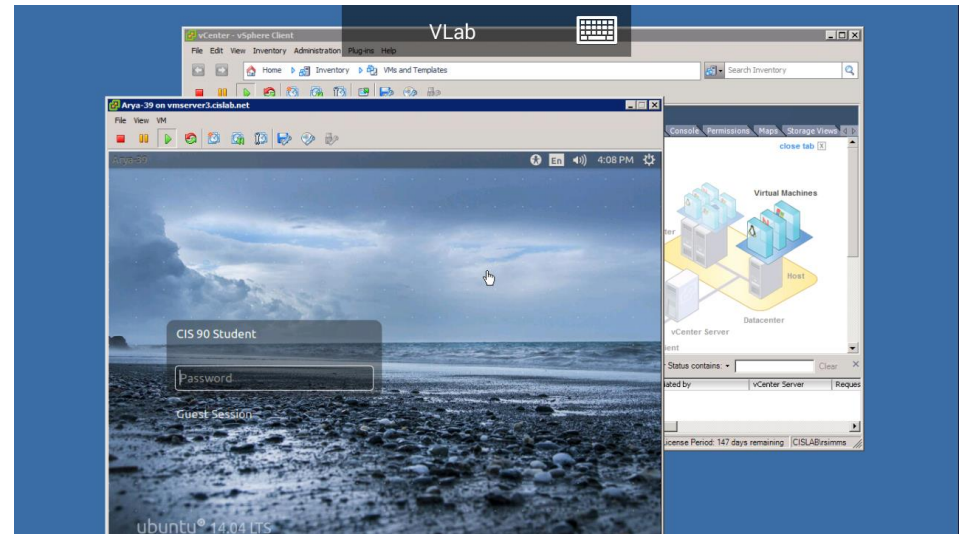
Android Terminal App



Viewing kernel version on
smartphone



Microsoft RDP App



Running Arya VM in VLab

Logging In

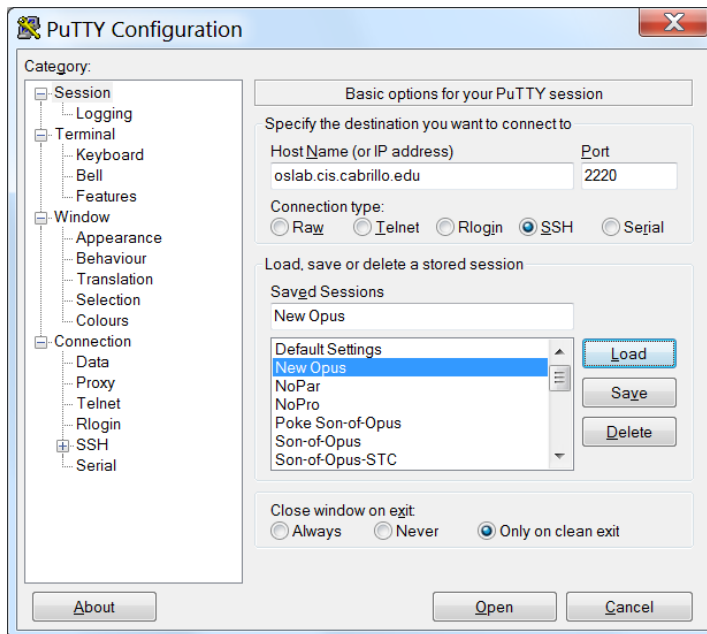


Who goes there?

What's the password?

<http://www.gutenberg.org/files/15064/15064-h/images/269.png>

Logging in



Use Putty to login from Windows PCs

Putty is a terminal emulator for Windows that supports the SSH protocol standard

```
ssh -p 2220 simben90@oslab.cis.cabrillo.edu
```

Use the ssh command to login from Macs or UNIX/Linux systems

Logging into Opus

username

For security reasons
the password is not
echoed as you type

Welcome to Opus
Serving Cabrillo College

Logging into Excalibur

username

- For security reasons the password is not echoed as you type

```
[simben90@excalibur ~]$
```

The /etc/passwd file stores all user accounts

cat /etc/passwd

```

Winter is coming

Last login: Sun Sep  7 17:20:31 2014 from opus.cis.cabrillo.edu
cis90@Arya-35:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin

```

snipped

```

whoopsie:x:109:116:/:nonexistent:/bin/false
speech-dispatcher:x:110:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/
sh
avahi:x:111:117:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
lightdm:x:112:118:Light Display Manager:/var/lib/lightdm:/bin/false
colord:x:113:121:colord colour management daemon,,,:/var/lib/colord:/bin/false
hplip:x:114:7:HPLIP system user,,,:/var/run/hplip:/bin/false
pulse:x:115:122:PulseAudio daemon,,,:/var/run/pulse:/bin/false
sshd:x:116:65534:/:/var/run/sshd:/usr/sbin/nologin
cis90:x:1000:190:CIS 90 Student:/home/cis90:/bin/bash
cis90@Arya-35:~$

```

Your username must match one of the accounts in the /etc/passwd file

Note: this file no longer contains the passwords!

The SUPER user

CIS 90 user

The /etc/shadow file stores the encrypted passwords

cat /etc/shadow

```
cis90@Arya-35: ~
cis90@Arya-35:~$ cat /etc/shadow
cat: /etc/shadow: Permission denied
cis90@Arya-35:~$ sudo cat /etc/shadow
[sudo] password for cis90:
root:$6$pVdCSve2$sYEEec2kIqgyXordPx17aEZCrR798kWxLuLmUD2Wr6oKnLUzVgV0cGce5CAw4wB
m3AhPE3XXDvkV40pDkuWlg0:16299:0:99999:7:::
daemon*:16177:0:99999:7:::
bin*:16177:0:99999:7:::
sys*:16177:0:99999:7:::
sync*:16177:0:99999:7:::
games*:16177:0:99999:7:::
man*:16177:0:99999:7:::
lp*:16177:0:99999:7:::
mail*:16177:0:99999:7:::
```

use sudo to run as
superuser (root)

The SUPER user

Your password must
match the account
password kept in
the /etc/shadow file

Only the root user
can view this file
and the passwords
are encrypted!

snipped

```
kernoops*:16177:0:99999:7:::
rtkit*:16177:0:99999:7:::
saned*:16177:0:99999:7:::
whoopsie*:16177:0:99999:7:::
speech-dispatcher:!:16177:0:99999:7:::
avahi*:16177:0:99999:7:::
lightdm*:16177:0:99999:7:::
colord*:16177:0:99999:7:::
hplip*:16177:0:99999:7:::
pulse*:16177:0:99999:7:::
sshd*:16230:0:99999:7:::
cis90:$6$TndkD0Zv$KMHSBc0AKCgrwAPXvPxKMmolRpaBcZFrPknxpv79xALYLlrZzJC9.6NLldzVX/
bd19XlQydsj3sp46L5cFS.O.:16299:0:99999:7:::
cis90@Arya-35:~$
```

CIS 90 user

Viewing your account in `/etc/passwd`

*This command, which we will learn how to do later, outputs **just one line** of the `/etc/passwd` file on Opus*

```
/home/cis90/simben $ grep simben90 /etc/passwd
```

```
simben90:x:1201:190:Benji Simms:/home/cis90/simben:/bin/bash
```

username

Comment

Home directory

Shell

Group ID (GID)

User ID (UID)

password (just a placeholder now)

Note the fields in `/etc/passwd` are delimited with a ":"

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

```
uid=1201(simben90) gid=190(cis90) groups=190(cis90),100(users)
```

```
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Now you know where the `id` command get some of its information!

Class Activity

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

username → **simben90**
password (just a placeholder now) → **x**
User ID (UID) → **1201**
Group ID (GID) → **190**
Comment → **Benji Simms**
Home directory → **/home/cis90/simben**
Shell → **/bin/bash**

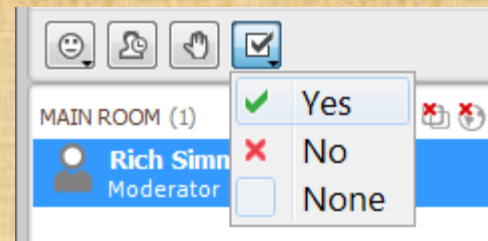
Note the field separator used in /etc/passwd is a ":"

1) Find your record in /etc/passwd

- Paste your UID (User ID) number in the chat window
- Paste your home directory in the chat window
- Paste your shell in the chat window

2) cat /etc/shadow

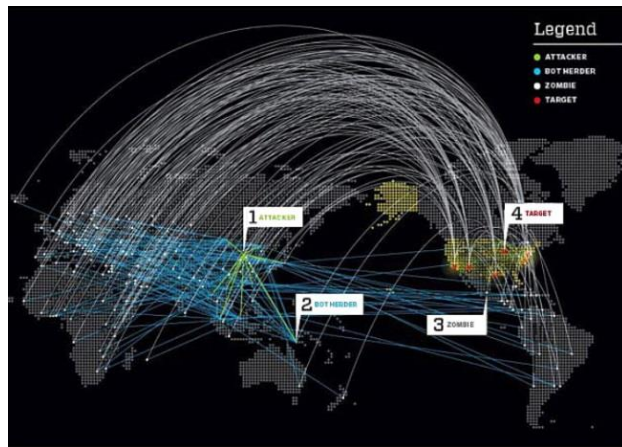
Give me a green check ✓ if you can view this file otherwise give me a red ✗



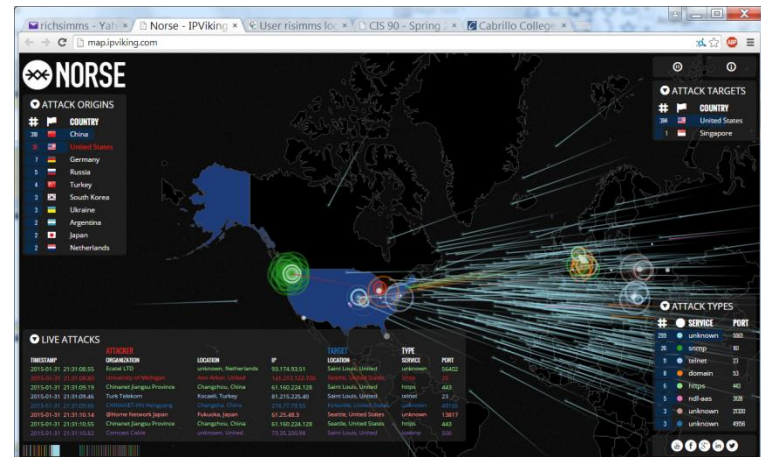
Passwords

Your password

- Strong passwords are critical!
- **Botnets** and malicious **ne-er-do-wells** are constantly attempting to break into computers attached to the Internet! (Even my little Frodo VM at home)



<http://mac-internet-security-software-review.toptenreviews.com/how-do-i-know-if-my-computer-is-a-botnet-zombie-.html>



<http://map.ipviking.com/>

They never stop trying

*The ne'er-do-wells trying to break in ...
this is why you need strong passwords*

----- SSHD Begin -----

```
SSHD Killed: 1 Time(s)
SSHD Started: 1 Time(s)
Disconnecting after too many authentication failures for user:
guest90 : 1 Time(s)
```

Failed logins from:

```
76.254.22.196 (adsl-76-254-22-196.dsl.pltn13.sbcglobal.net): 2 times
201.7.115.194 (201-7-115-194.spopa302.ipd.brasiltelecom.net.br): 2135 times
210.240.12.14: 20 times
```

Illegal users from:

```
201.7.115.194 (201-7-115-194.spopa302.ipd.brasiltelecom.net.br): 564 times
210.240.12.14: 42 times
```

```
Users logging in through sshd:
guest:
  76.254.22.196 (adsl-76-254-22-196.dsl.pltn13.sbcglobal.net): 2 times
jimg:
  70.132.20.25 (adsl-70-132-20-25.dsl.snfc21.sbcglobal.net): 7 times
ordazedw:
  76.254.22.196 (adsl-76-254-22-196.dsl.pltn13.sbcglobal.net): 1 time
root:
  63.249.86.11 (dsl-63-249-86-11.cruzio.com): 3 times
  70.132.20.25 (adsl-70-132-20-25.dsl.snfc21.sbcglobal.net): 1 time
rsimms:
  63.249.86.11 (dsl-63-249-86-11.cruzio.com): 2 times
```

From a logwatch report showing malicious attempts to break into Opus

/var/log/wtmp and var/log/btmp

```
[root@opus log]# lastb | sort | cut -f1 -d' ' | grep -v ^$ | uniq -c > bad
[root@opus log]# sort -g bad > bad.sort
[root@opus log]# cat bad.sort | tail -50
  471 ftp
  472 public
  490 test
  490 tomcat
  498 user
  506 service
  508 mike
  508 username
  524 cyrus
  530 pgsql
  532 test1
  544 master
  554 linux
  554 toor
  576 paul
  584 support
  590 testuser
  604 irc
  610 test
  656 noc
  686 www
  690 postfix
  723 john
  734 testing
  738 adam
  746 alex
  754 info
  798 tester
  832 library
  935 guest
  990 admin
 1002 office
 1022 temp
 1070 ftpuser
 1138 webadmin
 1298 nagios
 1332 web
 1374 a
 1384 student
 1416 postgres
 1690 user
 1858 oracle
 1944 mysql
 2086 webmaste
 5324 test
10803 root
10824 admin
18679 root
24064 root
[root@opus log]#
```

Top 50 usernames used by the ne'er-do-wells when attacking Opus

How to make a strong password

10

- The longer the better (~~8~~ or more characters)
- Not in any dictionary
- Use upper case, lowercase, punctuation, digits
- Something you can remember
- Keep it secret
- Change when compromised

Wh0le#!!

(Whole sh'bang)

KuKu4 (co) 2

(Cuckoo for Cocoa Puffs)

#0p&s@ve

(shop and save)

Idl02\$da

(I do laundry on Tuesday)

passwd command

Change user's password

Syntax:

passwd [username]

Example:

```
/home/cis90/simmsben $ passwd
Changing password for user simben90.
Changing password for simben90
(current) UNIX password: 
New UNIX password: 
Retype new UNIX password: 
passwd: all authentication tokens updated successfully.
/home/cis90/simmsben $
```

*Note, the passwords
are not echoed as
you type them.*

*This changes your password on Opus only (not
other VMs, the forum or BlackBoard)*

John the Ripper

An open source cracker that tries common passwords first followed by a brute force dictionary attack



Instructor: Use daughter and john-demo aliases to demo. Cat password.1st for common passwords.

For Supplemental Study

<https://www.grc.com/haystack.htm>

How Big is Your Haystack?
...and how well hidden is YOUR need?

Every password you use can be thought of as a needle hiding in a haystack. After all searches of common passwords and dictionaries have failed, an attacker must resort to a "brute force" search - ultimately trying every possible combination of letters, numbers and then symbols until the combination **you** chose, is discovered.

If every possible password is tried, sooner or later yours **will** be found.
The question is: Will that be **too** soon ... or **enough** later?

This interactive brute force search space calculator allows you to experiment with password length and composition to develop an accurate and quantified sense for the safety of using passwords that can only be found through exhaustive search. Please see the discussion below for additional information.

The Password Haystack Concept in 150 Seconds
Los Angeles' KABC-TV produced a terrific 1/2 hour and a half minute explanation of the Password Haystacks concept. Click this link to view their quick introduction.

GRC's Interactive Brute Force Password "Search Space" Calculator
(WARNING: you do have your browser's "What Happens Here, stays Here")

☒ No Uppercase ☒ Lowercase ☒ No Digits ☒ No Symbols ☒ 5 Characters

dummy

Enter and edit your test password in the field above while viewing the analysis below.

Brute Force Search Space Analysis:

Search Space Depth (Alphabet):	26
Search Space Length (Characters):	5 characters
Exact Search Space Size (Count): (count of all possible passwords with this alphabet size and up to the password's length)	12,356,630
Search Space Size (as a power of 10):	1.24 x 10 ⁷

Time Required to Exhaustively Search this Password's Space:

Online Attack Scenario: (Assuming one thousand guesses per second)	3.43 hours
Offline Fast Attack Scenario: (Assuming one hundred billion guesses per second)	0.000124 seconds
Massive Cracking Array Scenario: (Assuming one hundred trillion guesses per second)	0.000000124 seconds

Note that typical attacks will be online password guessing limited by, at most, a few hundred guesses per second.

(The Haystack Calculator has been viewed 2,145,143 times since its publication.)

ConsumerReports.org
The prestigious "Consumer Reports" has also picked up on the simplicity and power of the "Password Haystacks" concept.

IMPORTANT!!! What this calculator is NOT . . .
It is **NOT** a "Password Strength Meter."

Since it could be easily confused for one, it is very important for you to understand what it is, and what it isn't:
The #1 most commonly used password is "123456", and the 4th most common is "Password". So any password attacker and cracker would try those two passwords immediately. Yet the Search Space Calculator above shows the time to search for those two passwords online (assuming a very fast online rate of 1,000 guesses per second) at 18.52 minutes and 17.33 centuries respectively! If "123456" is the first password that's guessed, that wouldn't take 18.52 minutes. And no password cracker would wait 17.33 centuries before checking to see whether "Password" is the magic phrase.

Password strength calculator for random passwords

<https://www.youtube.com/watch?v=1ExUsGIfCrU>

Why Passwords Fail

- Unless people are using 10 character, completely random passwords, then their password isn't really good
- Example:
- pEl\NI{i8m
- If you make them use a password like that, they'll write it down
- Which also isn't good

CMPS 485: Password Complexity

Ryan Riley

Subscribe 24

88 views

Excellent presentation on making strong passwords



Housekeeping

Housekeeping

1. Send me your student survey
2. Lab 1 due by 11:59PM (Opus time) tonight

Use **submit** to turn in your work

Grading Rubric (30 points)

5 points for each correct scavenger hunt item

3 points - optional extra credit questions (1 point each).

Use **verify** to see what you turned in

3. Last day to drop/add is Saturday Feb 7th

Roll Call

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

Turn off recording

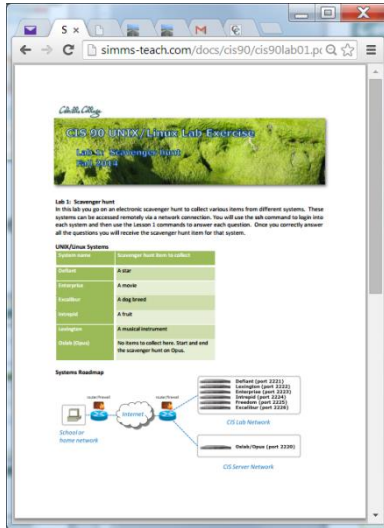
Do roll call using
both rosters

Turn on recording

Lab Assignments

Pearls of Wisdom:

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



Grading Code Names Lord of the Rings Characters

Current Progress					
Code Name	Grading Choice	Q1	Q2	Q3	Q4
Max Points		3	3	3	3
arwen	Grade				
arwen	Grade				
balin	Grade				
boromir	Grade				
denethor	Grade				
dwain	Grade				
elend	Grade				
eomer	Grade				
ewyn	Grade				
faramir	Grade				
frido	Grade				
galadriel	Grade				
gimli	Grade				
glorfindel	Grade				
leanna	Grade				
legolas	Grade				
luthien	Grade				
nazgul	Grade				
pippin	Grade				
saruman	Grade				
sauron	Grade				
theoden	Grade				
thranduil	Grade				

*I'll start sending out LOR code names this week for **everyone who sends or has sent me their survey.***

Introduction to UNIX/Linux (CIS 90)
Student Survey

Student Information

- Preferred first name: Last name:
- Date: Email address:
- Web site, if any:
- Grading choice: ☐ pass/no-pass ☐ grade (choose one, you may change your mind later)

Computer Background

- Previous computer classes or training taken:
- Work or other experience using computers:

Home equipment

- Do you have a working computer? ☐ yes ☐ no
- Operating system? ☐ Windows ☐ Mac ☐ Linux ☐ Other
- Internet connection? ☐ none ☐ dial-up ☐ dsl/cable

Course Objectives

- What are you hoping to learn in this class?
- Other comments or special learning needs?

(Please save & email completed survey to risimms@cabrillo.edu)

To get notifications of new 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".

The screenshot shows the phpBB forum interface for Cabrillo College: Computer and Information Systems. The forum is titled "CIS 90 - Spring 2015". It displays a list of topics with columns for replies, views, and last post. At the bottom of the forum, there is a "Board index" link and an "Unsubscribe forum" link. A callout box points to the "Unsubscribe forum" link.

TOPICS	REPLIES	VIEWS	LAST POST
The morning quiz by Ahmad Allulu » Thu Jan 29, 2015 4:11 pm	4	31	by Chris Copus » Sun Feb 01, 2015 12:21 pm
Lab 1 Bonus Question 2 clarification by Rich Simms » Sat Jan 31, 2015 5:33 pm	0	13	by Rich Simms » Sat Jan 31, 2015 5:33 pm
Bonus Question by Eduardo Fernandez » Thu Jan 29, 2015 2:02 pm	2	17	by Eduardo Fernandez » Fri Jan 30, 2015 9:21 pm
Lab 1 by Ryan Logan » Thu Jan 29, 2015 5:15 pm	0	18	by Ryan Logan » Thu Jan 29, 2015 5:15 pm
Test Test by Ahmad Allulu » Wed Jan 28, 2015 11:04 pm	1	22	by Michael Fegela » Thu Jan 29, 2015 9:59 am
Hello everyone by Chris Copus » Wed Jan 28, 2015 8:57 pm	1	19	by Michael Fegela » Thu Jan 29, 2015 9:58 am
CCC Conference on Ubuntu by Tim Converse » Wed Jan 28, 2015 6:05 pm	1	26	by Rich Simms » Wed Jan 28, 2015 9:12 pm
Hawknets wifi Login and Password by Ryan Logan » Wed Jan 28, 2015 4:02 pm	3	33	by Gerlinde Brady » Wed Jan 28, 2015 8:21 pm
Assignment turn in ? by mario perez » Wed Jan 28, 2015 3:46 pm	3	34	by mario perez » Wed Jan 28, 2015 3:58 pm

Display topics from previous: All Topics | Sort by Post time | Descending | Go

Mark topics read • 9 topics • Page 1 of 1

Return to Board Index

WHO IS ONLINE
Users browsing this forum: Rich Simms and 0 guests

FORUM PERMISSIONS
You can post new topics in this forum
You can reply to topics in this forum
You can edit your posts in this forum
You can delete your posts in this forum
You can post attachments in this forum

Board index | Unsubscribe forum | Contact us | The team | Members | Delete all board cookies • All times are UTC-08:00

Powered by phpBB® Forum Software © phpBB Limited
Administration Control Panel

Board index Unsubscribe forum

Help Available in the CIS Lab

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



Rich's Cabrillo College CIS Classes
CIS 90 Grades

Home

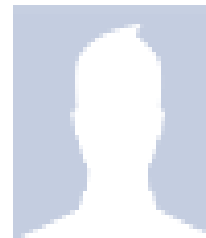
Resources

Forums

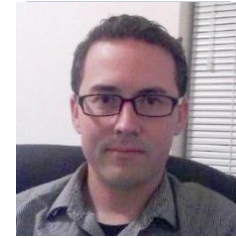
CIS Lab

Blackboard

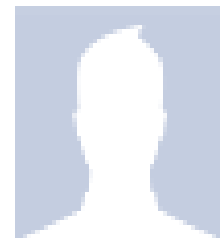
CIS 90 Student Lab Assistants:



Tess



Geoff

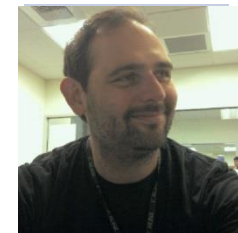


Nick

Linux Instructors



Rich Simms



Mike Matera

Look for Tess, Leandro, Nick, Rich or Mike on the schedule found [here](#)

CIS 90 Tutoring Available

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

TUTORIALS

ANNOUNCEMENTS & DEADLINES

- New subjects for Spring 2014:
- American Sign Language
- Computer Applications/Business Technology (CABT)
- Computer and Information Systems (CIS)
- History 17A

Welcome to the Tutorials Center!

We offer FREE peer tutoring to Cabrillo students who are enrolled in the course/s for which they need help.

- Tutoring is by appointment. The days and times of tutoring sessions are established by the office.
- Sessions are weekly and for the duration of the semester.
- Tutoring sessions are scheduled in small groups. Sessions last 1-2 hours depending on the class. Occasionally, sessions may be one to one but that is not guaranteed.
- Come directly to the TC office to schedule (second floor of library).

The following classes are being tutored for Spring 2014:

- Accounting 1A, 1B, 6, 54A, 151A, 159, 163
- American Sign Language (ASL) 1, 2
- Biology 4, 5, 6
- Computer Applications/Business Technology (CABT) 31, 38, 41, 101, 157, 160
- Computer and Information Systems (CIS) 81, 90, 172**
- Chemistry 1A, 1B, 2, 30A, 30B, 32

CONTACT INFORMATION

Tutorials Center

Location Room 1080A - Learning Resource Center

Phone 831.479.6470

Email tutorialcenter@cabrillo.edu

Coordinator Lori Chavez

Phone 831.479.6126

Email lochavez@cabrillo.edu

Hours Monday - Thursday: 9am - 5pm
Friday: 9am - 1pm

MAP, DIRECTIONS, & PARKING

DEPARTMENT STAFF & FACULTY DIRECTORY

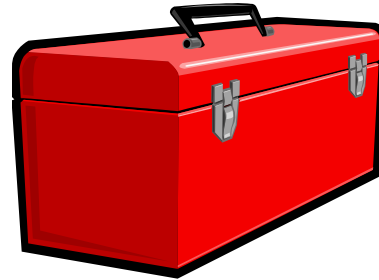


Matt Smithey

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

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

Don't wait too long to sign up! Tutoring hours are limited!



Lesson 2

Commands



Lesson 2 commands for your toolbox

echo
banner

Print text and variables

Make a banner

ls
cat
file
type
apropos
whatis
man
info

List directory contents

View file (name comes from concatenate)

Show additional information about a file

Shows where a command resides on the path

Searches the whatis database for strings

Searches the whatis database for commands

Show the manual page for a command

Alternate online documentation tool

bc
passwd

Binary calculator

Change password

set
env

List all shell variables

List all environment variables

echo command

Print text and variables

Syntax:

echo *[string]*

```
/home/cis90/simben $ echo hello rich  
hello rich
```

```
/home/cis90/simben $ echo joy to the world  
joy to the world
```

banner command

Output a banner

Syntax:

banner [string]

banner [string] [string] ... [string]

```
/home/cis90/simben $ banner I Love Linux
```

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

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

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

*Similar to echo command
but outputs banner sized
letters instead*

ls command

List files or directory contents

Syntax:

ls [pathname]

ls [pathname] [pathname] ... [pathname]

```
/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
```

*Listing the contents of
the current directory*

```
/home/cis90/simben $ ls Poems/
```

```
Angelou  Blake      Neruda     Shakespeare  Yeats
ant       Dickenson  nursery    twister
```

*Listing the contents of
the Poems directory*

```
/home/cis90/simben $ ls mission /bin/ps /usr/local/bin/banner
/bin/ps  mission  /usr/local/bin/banner
```

Listing three files

*Regular files show as black, directories show as blue and
executable programs/scripts show as green*

cat command

Concatenate and view file contents

Syntax:

cat *[pathname]*

cat *[pathname] [pathname] ... [pathname]*

```
/home/cis90/simben $ cat letter  
Hello Mother!  Hello Father!
```

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

< snipped >

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

type command

Locate a command on your path

Syntax:

type *[command]*

type *[command] [command] ... [command]*

```
[rsimms@opus run]$ type cal  
cal is /usr/bin/cal
```

*The **cal** command is located in the
/usr/bin directory*

```
/home/cis90/simben $ type bogus  
-bash: type: bogus: not found
```

*The **bogus** command is
not on the user's path*

```
[rsimms@opus run]$ type uname cal  
uname is /bin/uname  
cal is /usr/bin/cal
```

*The **uname** command is in the /bin directory
The **cal** command is in the /usr/bin directory*

name of the directory where file is found

name of the file (command/program)

Class Activity

1) Where is the **ssh** command?

Type your answer in the chat window.

file command

Show additional file information

Syntax:

file *[pathname]*

file *[pathname] [pathname] ... [pathname]*

```
/home/cis90/simben $ file letter  
letter: ASCII English text
```

```
/home/cis90/simben $ file Miscellaneous/  
Miscellaneous/: directory
```

```
/home/cis90/simben $ file timecal mission /usr/bin/cal  
timecal: Bourne-Again shell script text executable  
mission: ASCII English text  
/usr/bin/cal: ELF 32-bit LSB executable, Intel 80386, version 1  
(SYSV), dynamically linked (uses shared libs), for GNU/Linux  
2.6.18, stripped
```

Class Activity

1) Where is the **cat** command?

2) What kind of file is the **cat** command?

Type your answers in the chat window.

Answer:
The cat command is a type of
cat file command.
cat is a type of directory.
The cat command is the default
command for the cat command. It is used to
cat the cat command file.

apropos command

search the whatis database for strings

Syntax:

apropos *string*

```
/home/cis90/simben $ apropos echo
echo (1) - display a line of text
echo (1p) - write arguments to standard output
echo [builtins] (1) - bash built-in commands, see bash(1)
lessecho (1) - expand metacharacters
pam_echo (8) - PAM module for printing text messages
ping (8) - send ICMP ECHO_REQUEST to network hosts
ping6 [ping] (8) - send ICMP ECHO_REQUEST to network hosts
```


whatis command

search the whatis database for commands

Syntax:

whatis *command*

```
/home/cis90/simben $ whatis echo  
echo (1) - display a line of text  
echo (1p) - write arguments to standard output  
echo [builtins] (1) - bash built-in commands, see bash(1)
```

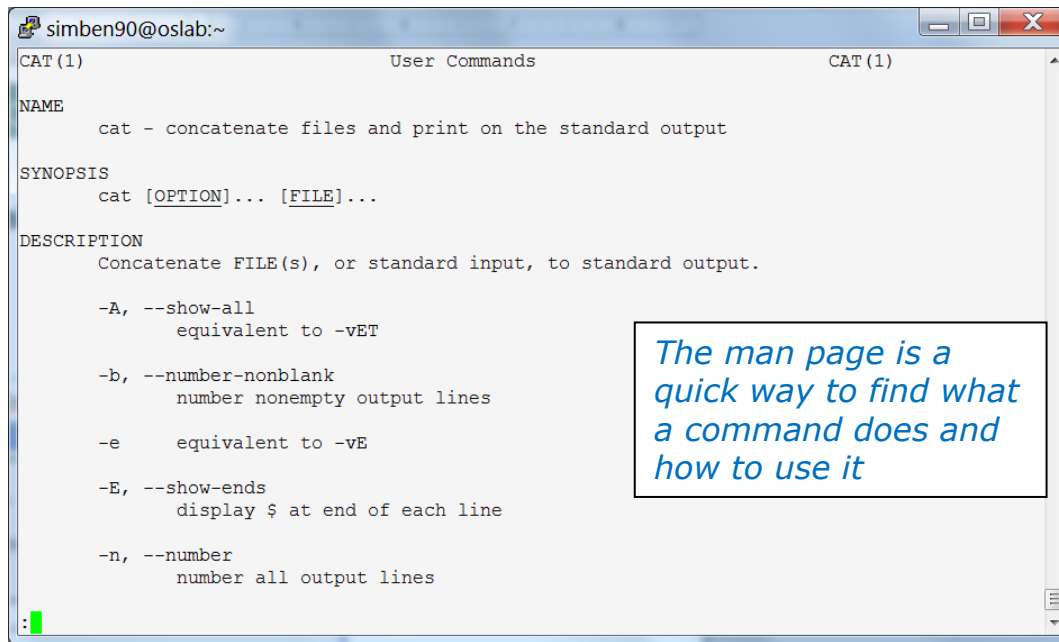
man command

Show the manual page (documentation) for a command

Syntax:

man *command*

```
/home/cis90/simben $ man cat
```



simben90@oslab:~

CAT (1) User Commands CAT (1)

NAME

cat - concatenate files and print on the standard output

SYNOPSIS

cat [OPTION]... [FILE]...

DESCRIPTION

Concatenate FILE(s), or standard input, to standard output.

-A, --show-all
equivalent to -vET

-b, --number-nonblank
number nonempty output lines

-e
equivalent to -vE

-E, --show-ends
display \$ at end of each line

-n, --number
number all output lines

: |

The man page is a quick way to find what a command does and how to use it



Use these keys to scroll



Use q key to quit

info command

Alternate documentation tool for commands

Syntax:

info command

Similar to man but has links to additional pages

/home/cis90/simben \$ **info** echo

```

simben90@oslab:~
features that only appear in Bash. Some of the shells that Bash has
borrowed concepts from are the Bourne Shell ('sh'), the Korn Shell
('ksh'), and the C-shell ('csh' and its successor, 'tcsh'). The
following menu k
one of these otl

This manual s
Bash. The Bash
on shell behavio

* Menu:
* Introduction::
* Definitions::
* Basic Shell Fe
* Shell Builtin
* Shell Variable
* Bash Features:
* Job Control::
* Command Line E
* Using History

--zz-Info: (bas

simben90@oslab:~
file: bash.info, Node: Shell Builtin Commands, Next: Shell Variables, Prev: Basic Shell\
Features, Up: Top
Shell Builtin Commands
*****
* Menu:
* Bourne Shell Builtin::      Builtin commands inherited from the Bourne
                              Shell.
* Bash Builtin::             Table of builtins specific to Bash.
* Modifying Shell Behavior:: Builtins to modify shell attributes and
                              optional behavior.
* Special Builtin::          Builtin commands classified specially by
                              POSIX.

Builtin commands are contained within the shell itself. When the
name of a builtin command is used as the first word of a simple command
(*note Simple Commands::), the shell executes the command directly,
without invoking another program. Builtin commands are necessary to
implement functionality impossible or inconvenient to obtain with
separate utilities.

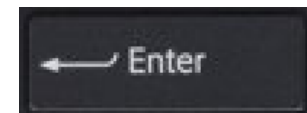
This section briefly describes the builtins which Bash inherits from
--zz-Info: (bash.info.gz)Shell Builtin Commands, 46 lines --Top-----
  
```



Use these keys to scroll



Use q key to quit



Use Enter to follow a link ()*



Use L to go back to last page

*Move cursor over an * and press Enter to follow link*

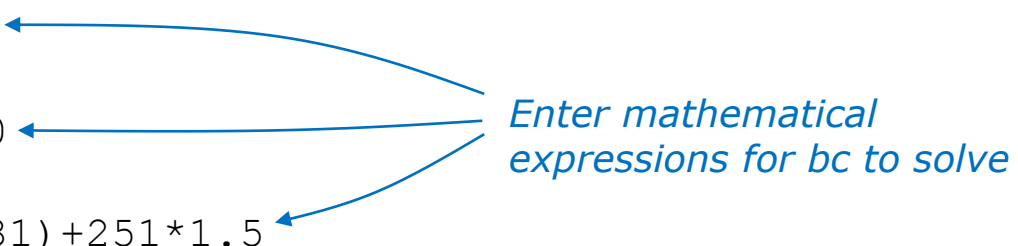
Class Activity

bc command

A binary calculator

Syntax:
bc

```
/home/cis90/simben $ bc
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006
Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
2+2
4
3*30
90
(3*31)+251*1.5
469.5
quit
/home/cis90/simben $
```



Enter mathematical expressions for bc to solve

*Use quit to
end program*

Class Activity

1) What does the following mathematical expression reduce to?

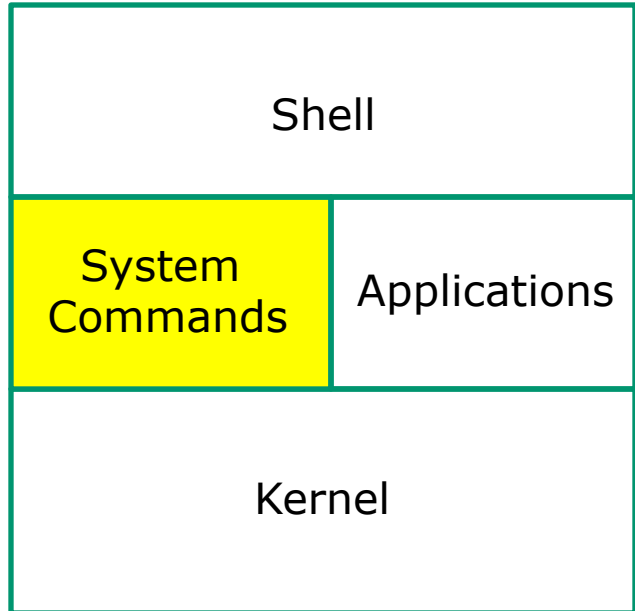
$$5342 * 56 - 2^5 - 299100 + (2 * 35)$$

Type your answer in the chat window

Where are the
UNIX/Linux
commands?

UNIX/Linux Architecture

System Commands



- 100's of system commands and utilities.
- Commands like **ls** (list directories), **cat** (print a file), **rm** (remove a file), ... etc.
- Utilities like **vi** (text editor), **sort** (sorts file contents), **find** (searches), ... etc.
- Larger utilities like **sendmail** (email), **tar** (backup), **tcpdump** (sniffer), ... etc.
- Administrative utilities like **useradd**, **groupadd**, **passwd** (change password), ... etc.



/bin

```

[raimma@server0-01 ~]$
arch      mut      fgrep      ls          pwd         sync
ash       date      gawk      mail        red         tar
ch_stario dd          grep      more       rm          tcat
awk       df         gtar      mkmdm      rsync       touch
username  dmesg      hostname  ntatmp     rpm         true
hash      find       ipsec     nmap       rvi         unalias
hash2     dmesg      hostname  mount      rview       uname
hash      domainname igmp       mt         sed         unlockd_start
ac         dumpsys    inotify    mv          sedfent     unlockd_stop
chgrp     echo      kill       nodestat   serial      unlink
shmdm     ed         kill      nice       sh          usleep
chown     egrep     link       nlsdman    sleep       vdir
cp         exeg      ln          psawk      sort        view
spio      mv        loadkeys  ping       stty        yfdomainname
ssh       cat        login     ps          su          zcat
[raimma@server0-01 ~]$

```

/sbin

[illegible]

/usr/bin

[illegible]

/usr/sbin

```

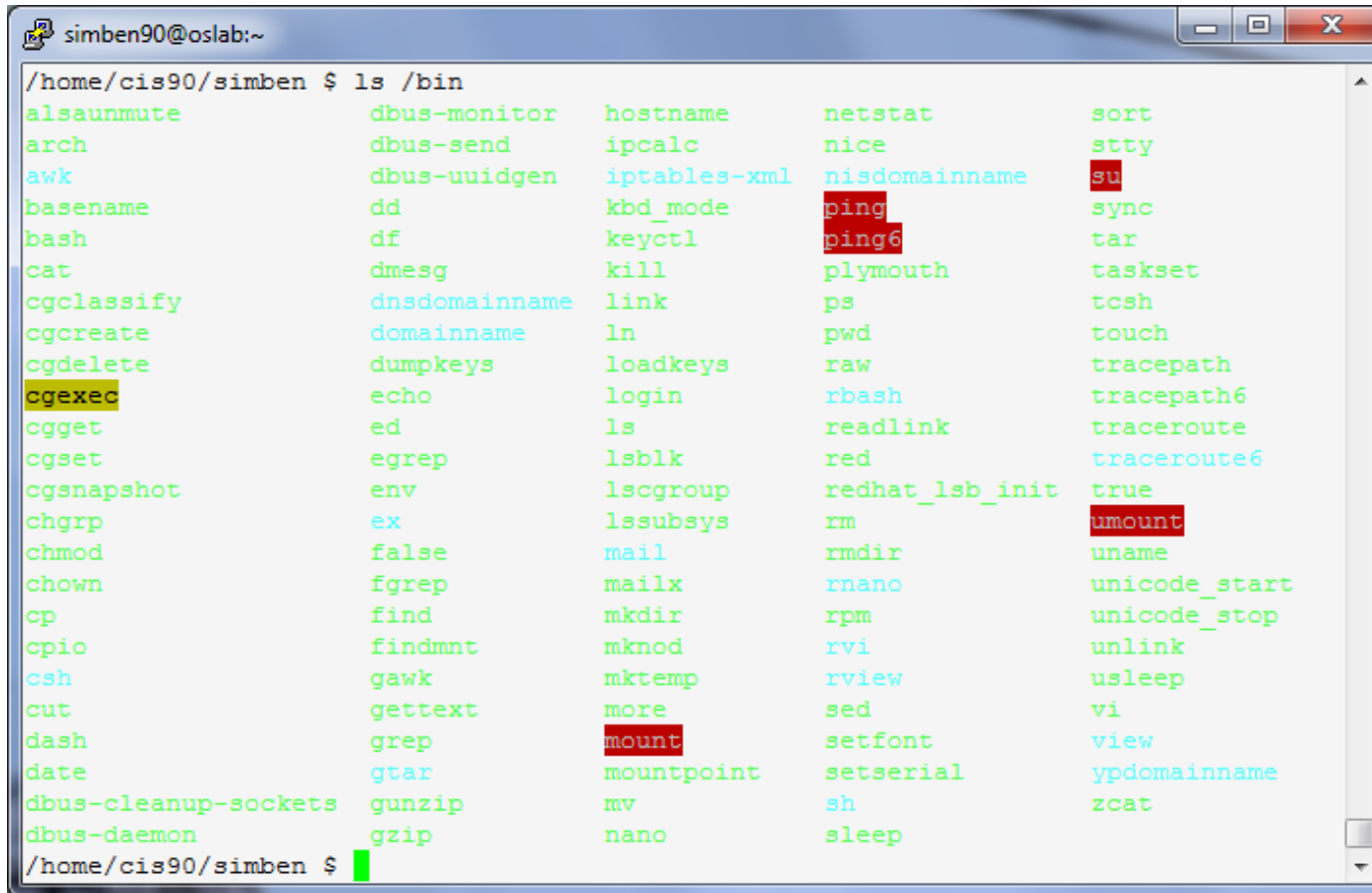
[isimms@server-01 isimms]$ ls /usr/sbin
accept                ntpd
adduser                nupdate
adsl-connect           nupdate
adsl-setup             ntp-generate
adsl-start             ntpd
adsl-status            ntpdate
adsl-stop              ntpdmonst
alternatives           ntptrace
anacron                ntp-wait
arpd                   ntpyes
arping                 packer
arpd                   pbsctool
atd                     blkpg
atrim                  blkpg
authconfig             pspp_dump
autocrypt              pspp_set
avahi-control          pspp
conobo-activation-sysoconf psppdump
build-locale-archive   psppoc
build-index-control    psppoc-relay
csmel-lock-helper      psppoc-server
cpmlnit                psppoc-sniff
cshfontpath            psppstat
                        psrliases

```

Most commands reside in these four directories. They can reside in other places as well.

The /bin directory

`ls /bin`



```

simben90@oslab:~
/home/cis90/simben $ ls /bin
alsaunmute      dbus-monitor    hostname        netstat         sort
arch            dbus-send       ipcalc         nice           stty
awk            dbus-uuidgen    iptables-xml   nisdomainname  su
basename       dd             kbd_mode       ping           sync
bash          df            keyctl         ping6          tar
cat           dmesg         kill          plymouth       taskset
cgclassify     dnsdomainname  link          ps            tcsh
cgcreate      domainname    ln           pwd          touch
cgdelete     dumpkeys     loadkeys      raw         tracepath
cgexec       echo         login        rbash        tracepath6
cgget        ed          ls          readlink     traceroute
cgset        egrep       lsblk       red          traceroute6
cgsnapshot   env         lscgroup    redhat_lsb_init true
chgrp        ex          lssubsys    rm           umount
chmod        false       mail        rmdir        uname
chown       fgrep       mailx       rnano        unicode_start
cp          find        mkdir       rpm          unicode_stop
cpio       findmnt    mknod      rvi          unlink
csh        gawk       mktemp     rview        usleep
cut        gettext   more       sed          vi
dash       grep      mount      setfont     view
date       gtar     mountpoint setserial  ypdomainname
dbus-clean-up-sockets gunzip mv sh zcat
dbus-daemon gzip    nano  sleep
/home/cis90/simben $
  
```

/bin has essential commands used by everyone.

*Can you find the Lesson 1 **date**, **hostname**, **ps** and **uname** commands?*

*Can you find the **bash** shell?*

Commands are either program or script files that can be executed

The /usr/bin directory

`ls /usr/bin`

```
simben90@oslab:~
/home/cis90/simben $ ls /usr/bin
[
a2p                gst-feedback-0.10    powertop
ab                 gst-inspect          ppdc
abrt-action-analyze-backtrace  gst-inspect-0.10    ppdhtml
abrt-action-analyze-c         gst-launch            ppdi
abrt-action-analyze-core      gst-launch-0.10      ppdmerge
abrt-action-analyze-oops      gst-typefind          pppdo
abrt-action-analyze-python    gst-typefind-0.10    ppl-config
abrt-action-generate-backtrace  gst-xmlinspect       ppm2tiff
abrt-action-install-debuginfo  gst-xmlinspect-0.10  pr
abrt-action-list-dsos          gst-xmllaunch         precat
abrt-action-save-package-data  gst-xmllaunch-0.10   pre-grohtml
abrt-action-trim-files         gtbl                  preunzip
abrt-cli                   gtk-query-immodules-2.0-32  prezip
abrt-dump-oops              gtk-update-icon-cache  prezip-bin
                           gtroff                printafm
```

There are a "ton" of additional commands (programs) in this directory.

You will need to scroll through a lot of pages to see them all!

snipped

```
grotty              png2theora           zforce
groups              pnm2ppa              zgrep
gs                  pod2html              zip
gsbj                pod2latex             zipcloak
gsdj                pod2man               zipgrep
gsdj500             pod2text              zipinfo
gslj                pod2usage             zipnote
gslp                podchecker            zipsplit
gsnd                podselect             zless
gsocelim            POST                  zmore
gstack              post-grohtml          znew
gst-feedback         poweroff              zsoelim
```

*Can you find the Lesson 1 **cal**, **clear**, **id**, **ssh**, **tty**, and **who** commands we used in Lab 1?*

The /sbin directory

```
ls /sbin
```

```

/home/cis90/simben $ ls /sbin
accton      fsck.cramfs  kpartx      nameif       scsi_id
addpart     fsck.ext2    ldconfig    netreport    securetty
agetty      fsck.ext3    load_policy new-kernel-pkg service
alsactl     fsck.ext4    logsave     nologin      setpci
arp         fsck.ext4dev losetup      pam_console_apply setregdomain
arping      fsck.msdos   lsinitrd    pam_tally2   setregdomain
audispd     fsck.vfat    lsmod       pam_timestamp_check setsysfont
auditctl    fsfreeze     lspci       parted       sfdisk
auditd      fstab-decode lspcmcia     partprobe    sgpio
aureport    fstrim       lvchange     partx        shutdown
ausearch    fuser        lvconvert    pccardctl    slattach
autrace     genhostid    lvcreate     pidof        sln
badblocks   getkey       lvdisplay    pivot_root   start
blkid       grub         lvextend     plipconfig   start_udev
blockdev    grubby      lum          plumouthd    status

```

snipped

```

dumpe2fs    iptables-restore mkfs.ext4      restorecon    vgimport
e2fsck      iptables-save   mkfs.ext4dev  rfkill        vgimportclone
e2image     iptunnel        mkfs.msdos    rmmod         vgmerge
e2label     iw              mkfs.vfat     rmt           vgmknodes
e2undo      iwconfig        mkhomedir_helper rngd          vgreduce
ether-wake  iwevent         mkinitrd      route         vgrename
ethtool     iwgetid         mkswap        rpcbind       vgrename
faillock    iwlist          modinfo       rpc.statd     vgs
fdisk       iwpriv          modprobe      rrestore      vgscan
findfs      iwspy           mount.cifs    rsyslogd     vgsplit
fixfiles    kdump           mount.nfs     rtmon         weak-modules
fsadm       kexec           mount.nfs4    runlevel      wipefs
fsck        killall5        mount.tmpfs   runuser

```

```

/home/cis90/simben $

```

These are essential commands and utilities used by system administrators.

*This is where the **chkconfig**, **ifconfig** and **iptables** commands are found.*

You will learn how to use these commands in CIS 191 and CIS 192.

The /usr/sbin directory

```
ls /usr/sbin
```

```

simben90@oslab:~/
/home/cis90/simben $ ls /usr/sbin
abrttd                                hald                                  pwconv
abrt-install-ccpp-hook                htcacheclean                         pwunconv
abrt-server                            httpd                                 quota_nld
accept                                 httpd.event                           quotastats
accton                                 httpd.worker                          raid-check
acpid                                  httxt2dbm                             readprofile
addgnupghome                           hwclock                               redhat_lsb_trigger.i686
adduser                                iconvconfig                           reject
alsactl                                iconvconfig.i686                     repquota
alternatives                           ipa-client-install                   restorecond
anacron                                ipa-getkeytab                         rotatelog
apachectl                              ipa-join                              rpcdebug
applygnupgdefaults                    ipa-rmkeytab                         rpc.gssd
arpd                                    irqbalance                           rpc.idmapd
arping                                 kbd5_send_err                        xzinfo
snipped
getenforce                             postconf                             userhelper
getpcaps                               postdrop                             usermod
getsebool                              postfix                              usernetctl
glibc_post_upgrade.i686                postkick                             vipw
groupadd                               postlock                             visudo
groupdel                               postlog                              vpddecode
groupmems                              postmap                              vsftpd
groupmod                               postmulti                            warnquota
grpck                                  postqueue                           yum-complete-transaction
grpconv                               postsuper                           yumdb
grpunconv                             praliases                           zdump
gss_clnt_send_err                     prelink                              zic
gss_destroy_creds                      pwck
/home/cis90/simben $

```

These are additional commands and utilities are typically used by system administrators.

*This is where
commands like
useradd,
userdel,
tcpdump are
located.*

You will learn how to use these commands in CIS 191 and CIS 192.

type command

Locate a command on your path

Syntax:

type *[command]*

type *[command] [command] ... [command]*

```
[rsimms@opus run]$ type cal  
cal is /usr/bin/cal
```

*The **cal** command is located in the
/usr/bin directory*

```
/home/cis90/simben $ type bogus  
-bash: type: bogus: not found
```

*The **bogus** command is
not on the user's path*

```
[rsimms@opus run]$ type uname cal  
uname is /bin/uname  
cal is /usr/bin/cal
```

*The **uname** command is in the /bin directory
The **cal** command is in the /usr/bin directory*

name of the file (command/program)

name of the directory where file is found

Class Activity

Draw a line connecting the command to the directory where it resides

bc

/bin

tty

/usr/bin

echo

/sbin

ifconfig

/usr/sbin

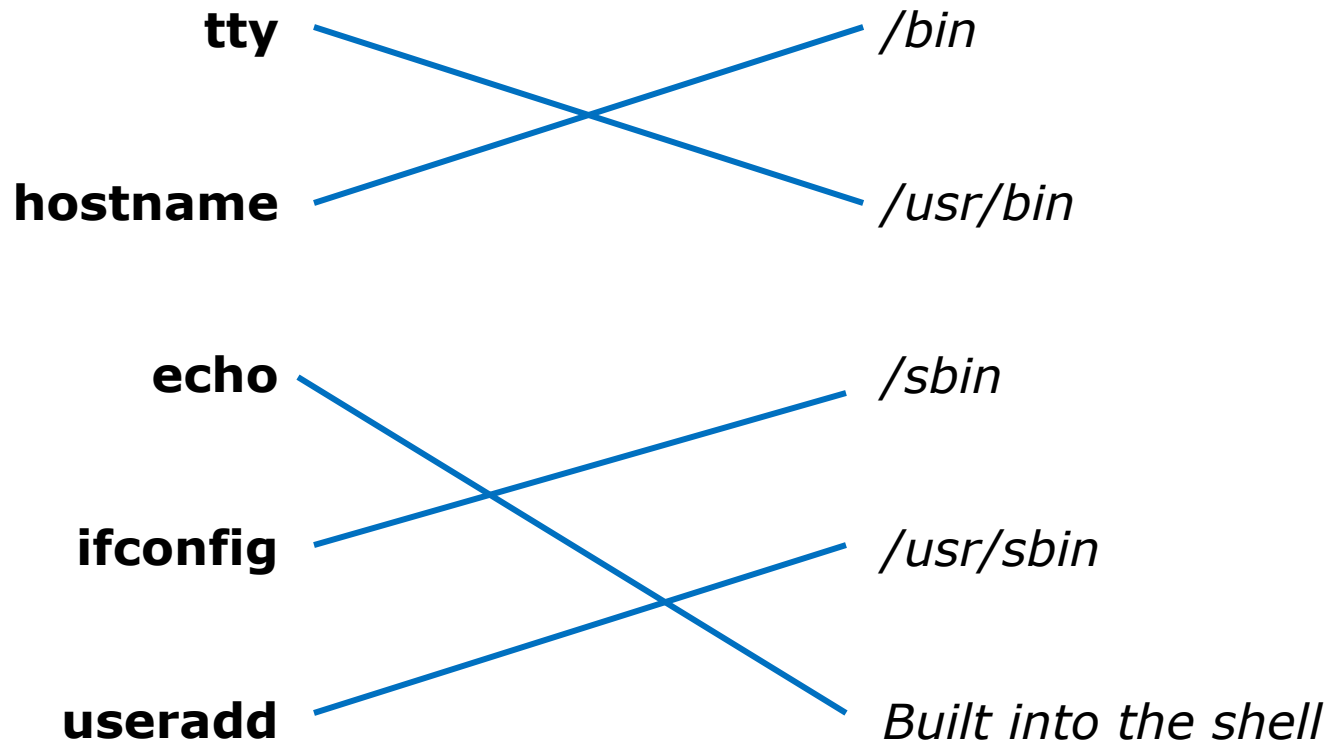
useradd

Built into the shell

You will learn more about useradd in CIS 191 and ifconfig in CIS 192

Class Activity

Draw a line connecting the command to the directory where it resides





Programs

Binary code
vs text scripts

UNIX commands & utilities are executable programs

A program can be binary code:

- Binary machine code is unprintable. A programmer must use hex dumps to examine it.
- Binary machine code executes very quickly and is targeted for a specific CPU instruction set.
- The binaries are produced by compiling source code written in a higher level language such as C, or C++.

A program can be a text-based script:

- A script can be directly viewed and printed.
- A script does not need to be compiled. It is interpreted on the fly and because of that doesn't run as fast as binary code.
- Common scripting languages include bash, perl and python.

Two example programs: apropos and cal

Lets take a deep dive on two random commands:

apropos - searches the whatis database for a string of text

cal - prints a calendar

*I'll be using this graphic to indicate
a program that has been loaded
into memory to be executed*





apropos

What do they do?



cal

*The **apropos** command searches the *whatis* database.*

```
/home/cis90/simben $ apropos uname
oldolduname [obsolete] (2) - obsolete system calls
olduname [obsolete] (2) - obsolete system calls
uname (1) - print system information
uname (1p) - return system name
uname (2) - get name and information about current kernel
uname (3p) - get the name of the current system
```

*The **cal** command prints a calendar*

```
/home/cis90/simben $ cal
February 2012
Su Mo Tu We Th Fr Sa
      1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29
```

Where are the programs located?



apropos



cal

```
/home/cis90/simben $ type apropos cal  
apropos is hashed (/usr/bin/apropos)  
cal is /usr/bin/cal
```

The **apropos** and **cal** commands are both in the **/usr/bin** directory.

Note: Sometimes you will see "Hashed" which means the command has been run previously and its location on the path has been temporarily "remembered". This is to speed up subsequent path searches for the same command.

Listing the program files



apropos



cal

```
/home/cis90/simben $ ls /usr/bin/apropos /usr/bin/cal  
/usr/bin/apropos  /usr/bin/cal
```

*Both files show as green
because they are
executables*

```
/home/cis90/simben $ ls -F /usr/bin/apropos /usr/bin/cal  
/usr/bin/apropos*  /usr/bin/cal*
```

*FYI, use the -F option if
color blind. Executables
have a * suffix.*

Getting additional information on the program files



apropos



cal

```
/usr/bin $ file apropos  
apropos: Bourne shell script text executable
```

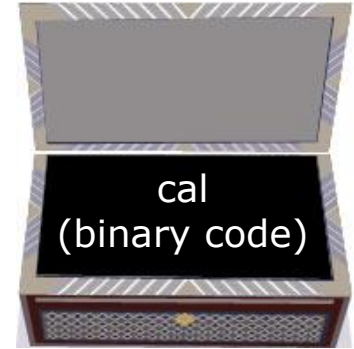
***apropos** is a
shell script*

```
/usr/bin $ file cal  
cal: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),  
for GNU/Linux 2.6.9, dynamically linked (uses shared libs),  
for GNU/Linux 2.6.9, stripped
```

***cal** is binary code (has been compiled
from higher level source code)*



Viewing the contents of the program files



```
cat /usr/bin/apropos
```

```
simben90@oslab:~  
/home/cis90/simben $ cat /usr/bin/apropos  
#!/bin/sh  
#  
# apropos -- search the whatis database for keywords.  
# whatis -- idem, but match only commands (as whole words).  
#  
# Copyright (c) 1990, 1991, John W. Eaton.  
# Copyright (c) 1994-1999, Andries E. Brouwer.  
#  
# You may distribute under the terms of the GNU General Public  
# License as specified in the README file that comes with the man  
# distribution.  
#  
# apropos/whatis-1.5m aeb 2003-08-01 (from man-1.6f)  
#  
# keep old PATH - 000323 - Bryan Henderson  
# also look in /var/cache/man - 030801 - aeb  
  
program=`basename $0`  
  
# When man pages in your favorite locale look to grep like binary files  
# (and you use GNU grep) you may want to add the 'a' option to *grepopt1.  
aproposgrept1='ai'  
aproposgrept2=''  
whatisgrept1='aiw'  
whatisgrept2='^'
```

The **cat** command can print the apropos file because it is a readable (and editable) **ASCII** text script

```
cat /usr/bin/cal
```

[illegible]

The **cat** command "chokes" trying to print the **binary** cal file because it is full of unprintable machine code.

How binary programs are created



cal

From: gcal-3.01.tar.gz

```
rsimms@nosmo:~/depot/gcal-3.01/src
[rsimms@nosmo src]$ head -50 gcal.c
/*
 * gcal.c: Main part which controls the extended calendar program.
 *
 * Copyright (c) 1994, 95, 96, 1997, 2000 Thomas Esken
 *
 * This software doesn't claim completeness, correctness or usability.
 * On principle I will not be liable for ANY damages or losses (implicit
 * or explicit), which result from using or handling my software.
 * If you use this software, you agree without any exception to this
 * agreement, which binds you LEGALLY !!
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the 'GNU General Public License' as published by
 * the 'Free Software Foundation'; either version 2, or (at your option)
 * any later version.
 *
 * You should have received a copy of the 'GNU General Public License'
 * along with this program; if not, write to the:
 *
 * Free Software Foundation, Inc.
 * 59 Temple Place - Suite 330
 * Boston, MA 02111-1307, USA
 */

static char rcsid[]="$Id: gcal.c,v 1.1 1994/01/01 00:00:00, Thomas Esken$";

/*
 * Include header files.
 */
#include "tailor.h"
#ifdef HAVE_ASSERT_H
#include <assert.h>
#endif
#ifdef HAVE_CTYPE_H
#include <ctype.h>
#endif
```

*Note: The **cal** binary code resulted from compiling the original gcal.c source code.*

```
rsimms@nosmo:~/depot/gcal-3.01/src
[rsimms@nosmo src]$ file /usr/bin/cal
/usr/bin/cal: ELF 32-bit LSB executable, Intel 80386, version 1
(SYSV), for GNU/Linux 2.2.5, dynamically linked (uses shared lib
s), stripped
[rsimms@nosmo src]$
```

Because GNU Linux software is licensed under the GPL you can make your own custom version of the commands or the kernel!

FYI

See this forum post from a previous class for an example of obtaining the source code for a Linux command and modifying it:

<http://oslab.cabrillo.edu/forum/viewtopic.php?f=31&t=683&p=2774>

Lab #2...even though 'info uname' output states...

by Dan McNamara » Fri Feb 18, 2011 12:53 pm

Hi Folks,

Does anyone happen to know if there are ways to manipulate output from uname such that it is listed in the order that I want it to be? Under 'Commands' in Lab #2, question 11, we are asked what options would we use to display just the operating system, its kernel release numbers and the machine's network node hostname. I got that okay. However, what if I wanted the output to display following the constructs of the question, i.e.:

```
opus.cabrillo.edu 2.6.18-164.el5 GNU/Linux (the default)
```

```
GNU/Linux 2.6.18-164.el5 opus.cabrillo.edu (what I'd like it to be)
```

Doing a 'man uname' doesn't cover this but 'info uname' states:

If multiple options or '-a' are given, the selected information is printed in this order:

```
KERNEL-NAME NODENAME KERNEL-RELEASE KERNEL-VERSION  
MACHINE PROCESSOR HARDWARE-PLATFORM OPERATING-SYSTEM
```

I can live with the default output as it does answer the question...it just kind of bugs me that it's not in the order that I would prefer. Mixing the order of the options has no effect on the default output.

Just wondering....



Dan McNamara

Posts: 38
Joined: Fri Feb 04, 2011 5:21 pm

*It all started when Dan did Lab 2 and wanted to change the way **uname** ordered its output!*

Class Activity

- 1) Where is the **scavenge** program?

Hint: use the **type** command with `scavenge` as the argument.

Type your answer in the chat window.

- 2) Is the **scavenge** command a binary executable or a shell script?

Hint: use the **file** command with the location of `scavenge` as the argument.

Type your answer in the chat window.

- 3) Can you **cat** the **scavenge** command?

Paste a line of output in the chat window.

- 4) Is **scavenge** a UNIX command?

Hint: use the **man** or **whatis** commands with `scavenge` as the argument.

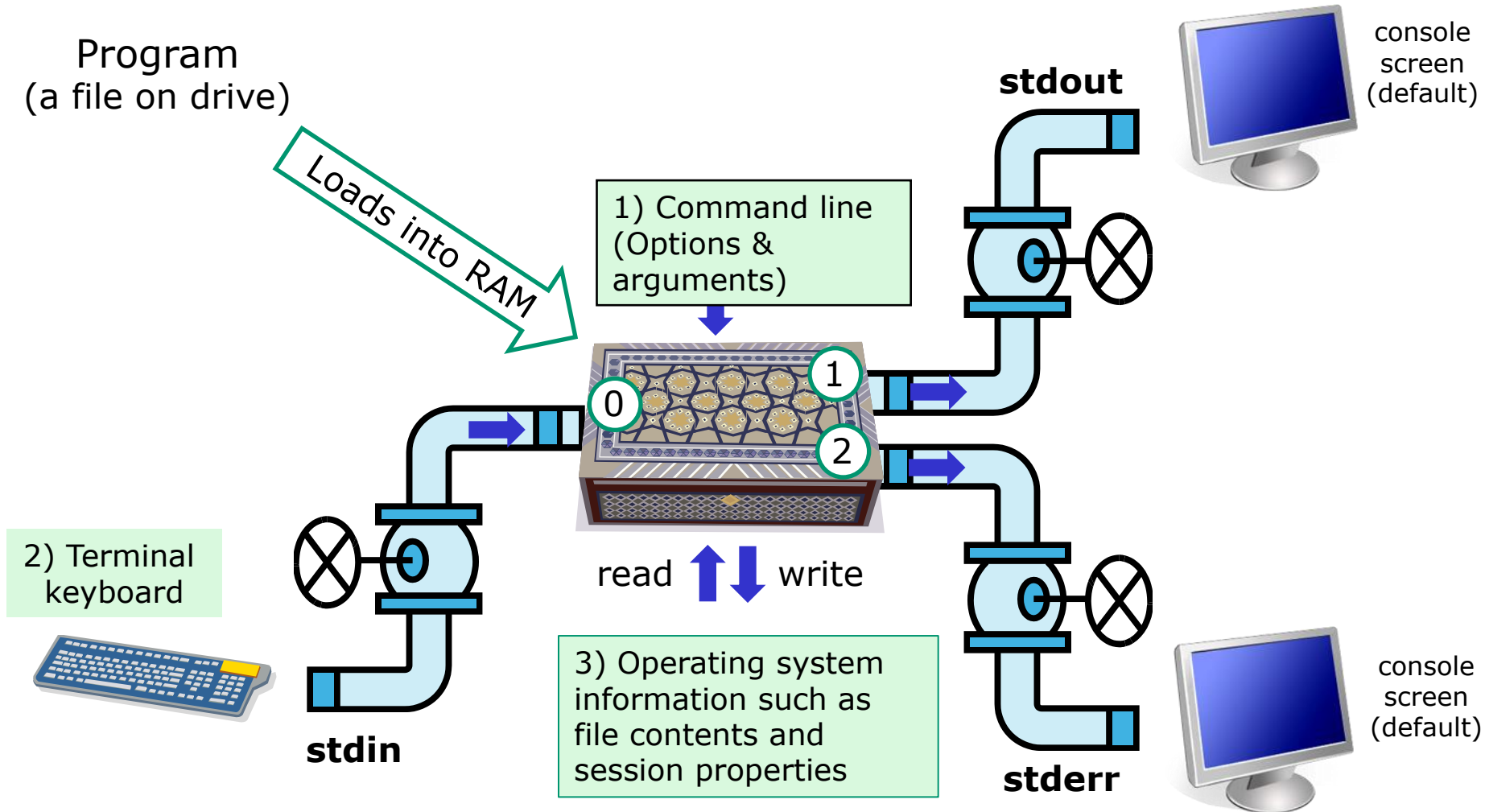
Type your answer in the chat window.

Inputs to commands

You will get these questions when you submit Lab 2

- 1) Name a UNIX command that gets its input only from the command line?
- 2) Name an interactive command that reads its input from the keyboard?
- 3) Name a UNIX command that gets its input from the Operating System?

Inputs to Commands



Name a UNIX command that gets its input only from the command line?

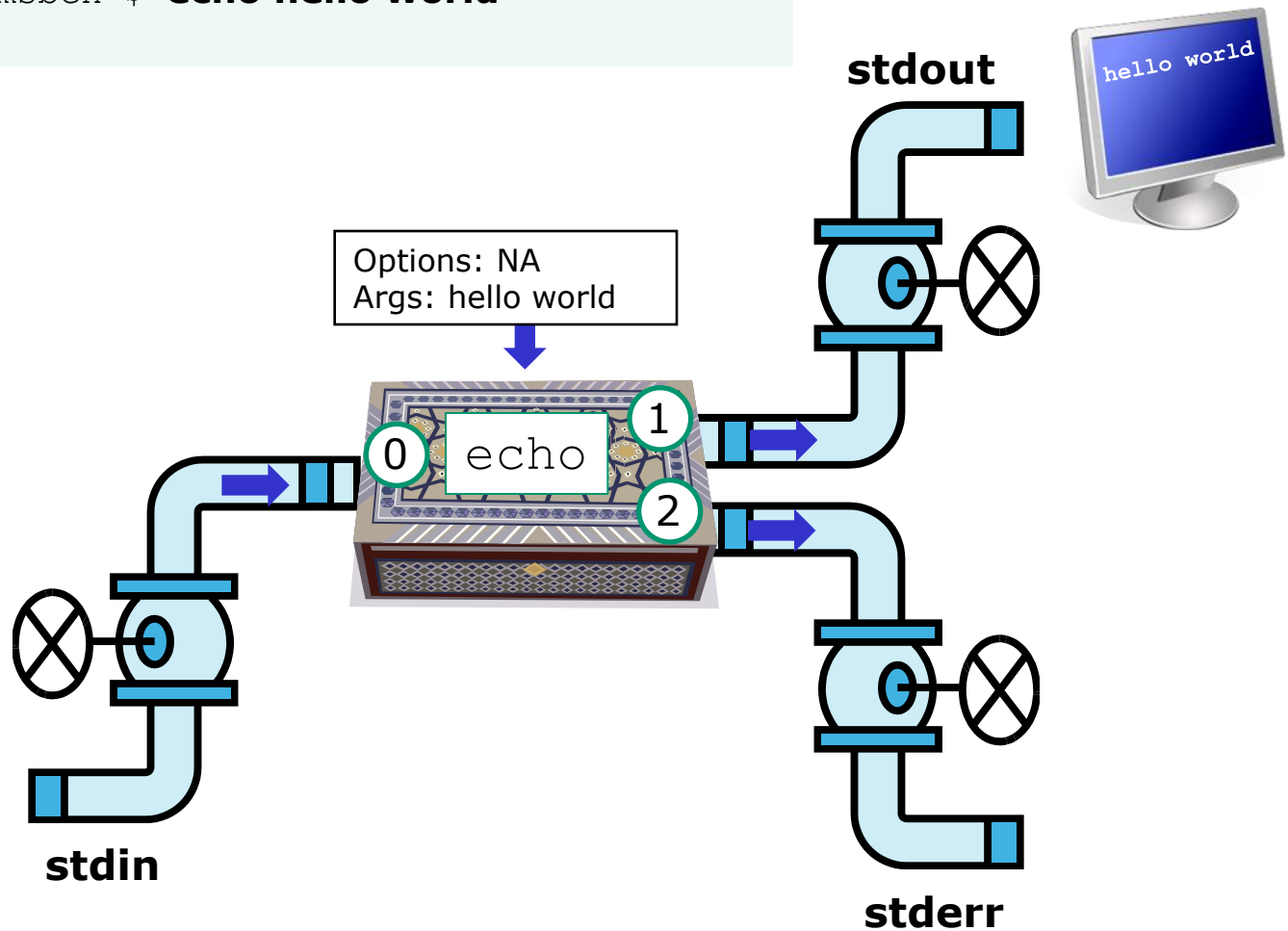
```
/home/cis90/simmen $ echo hello world  
hello world
```

```
/home/cis90/simben $ banner hello world  
#      # ##### #      #      #####  
#      # #      #      #      #  
#      # #      #      #      #  
##### ##### #      #      #  
#      # #      #      #      #  
#      # #      #      #      #  
#      # ##### ##### #####  
  
#      # ##### ##### #      #####  
# # # #      # #      # #      #  
# # # #      # #      # #      #  
# # # #      # ##### #      #  
# # # #      # #      # #      #  
# # # #      # #      # #      #  
## ## ##### #      # #####
```

*The **echo** and **banner** commands are examples of commands that get their input from the command line*

echo command

```
/home/cis90/simmsben $ echo hello world  
hello world
```



*The **echo** command is an example of a command that gets its input from the command line*

Name an interactive command that reads its input from the keyboard?

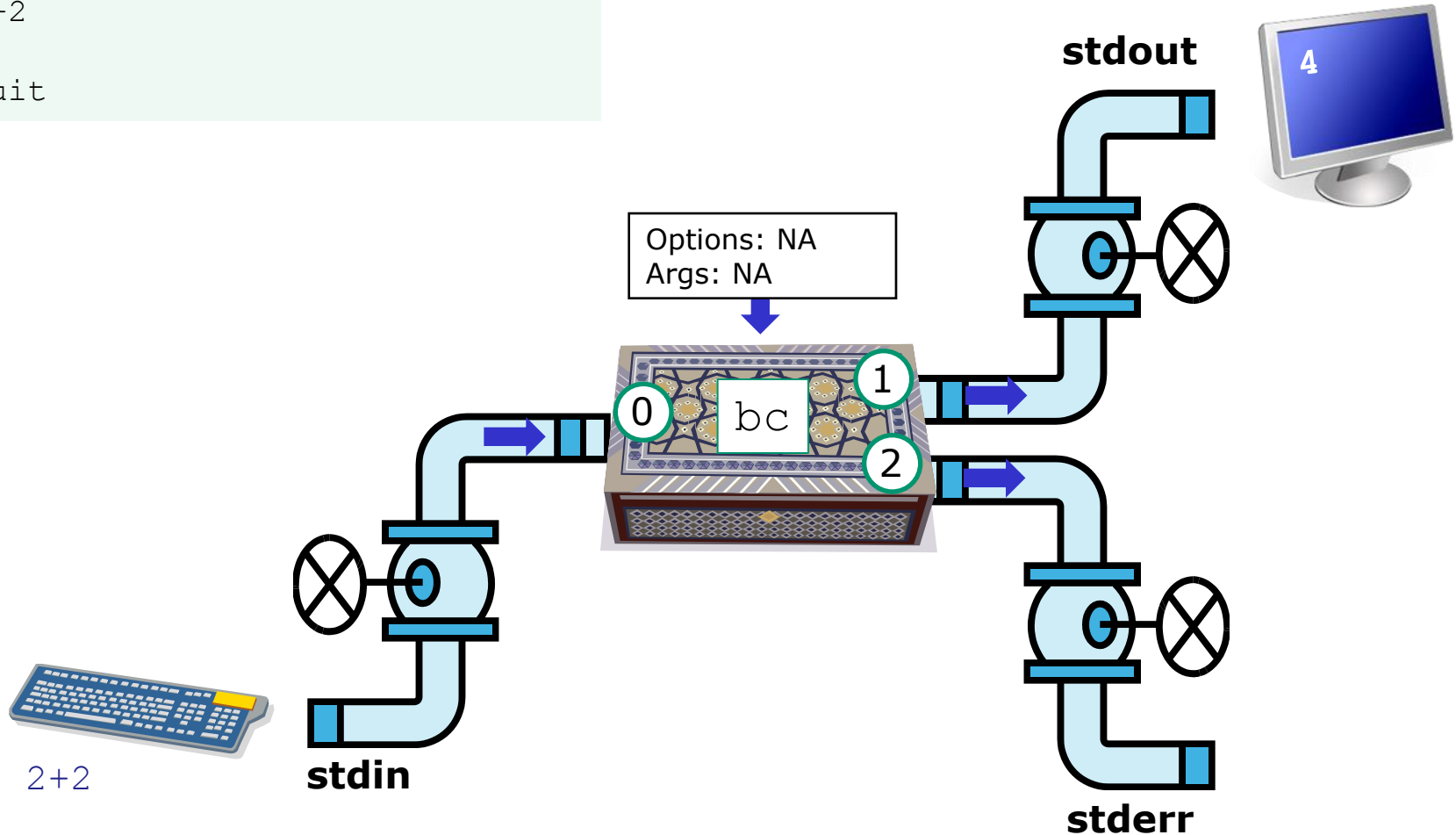
```
/home/cis90/simmsben $ bc
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free
Software Foundation, Inc.
This is free software with ABSOLUTELY NO
WARRANTY.
For details type `warranty'.
2+2
4
500-200+3
303
sqrt(64)
8
quit
```

```
/home/cis90/simmsben $ passwd
Changing password for user simmsben.
Changing password for simmsben
(current) UNIX password:
New UNIX password:
BAD PASSWORD: is too similar to the old
one
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated
successfully.
```

*The **bc** (binary calculator) and **passwd** commands are examples of interactive commands that read their input from the keyboard*

bc command

```
[rsimms@nosmo ~]$ bc
<snipped>
2+2
4
quit
```



*The **bc** (binary calculator) command is an example of an interactive command that reads its input from the keyboard*

Name a **UNIX** command that gets its input from the Operating System?

```
/home/cis90/simben $ who
dycktim pts/1      2010-09-07 17:07 (nosmo-nat.cabrillo.edu)
root    :0          2009-12-18 17:30
velasoli pts/2      2010-09-07 17:08 (adsl-35-201-114-102.dsl.net)
guest90 pts/3      2010-09-07 16:56 (nosmo-nat.cabrillo.edu)
rsimms  pts/4      2010-09-07 15:54 (dsl-45-78-13-81.dhcp.com)
guest90 pts/5      2010-09-07 16:59 (nosmo-nat.cabrillo.edu)
watsohar pts/6      2010-09-07 17:03 (nosmo-nat.cabrillo.edu)
swansgre pts/7      2010-09-07 17:10 (nosmo-nat.cabrillo.edu)
guest90 pts/8      2010-09-07 17:10 (nosmo-nat.cabrillo.edu)
abbenste pts/9      2010-09-07 17:11 (nosmo-nat.cabrillo.edu)
```

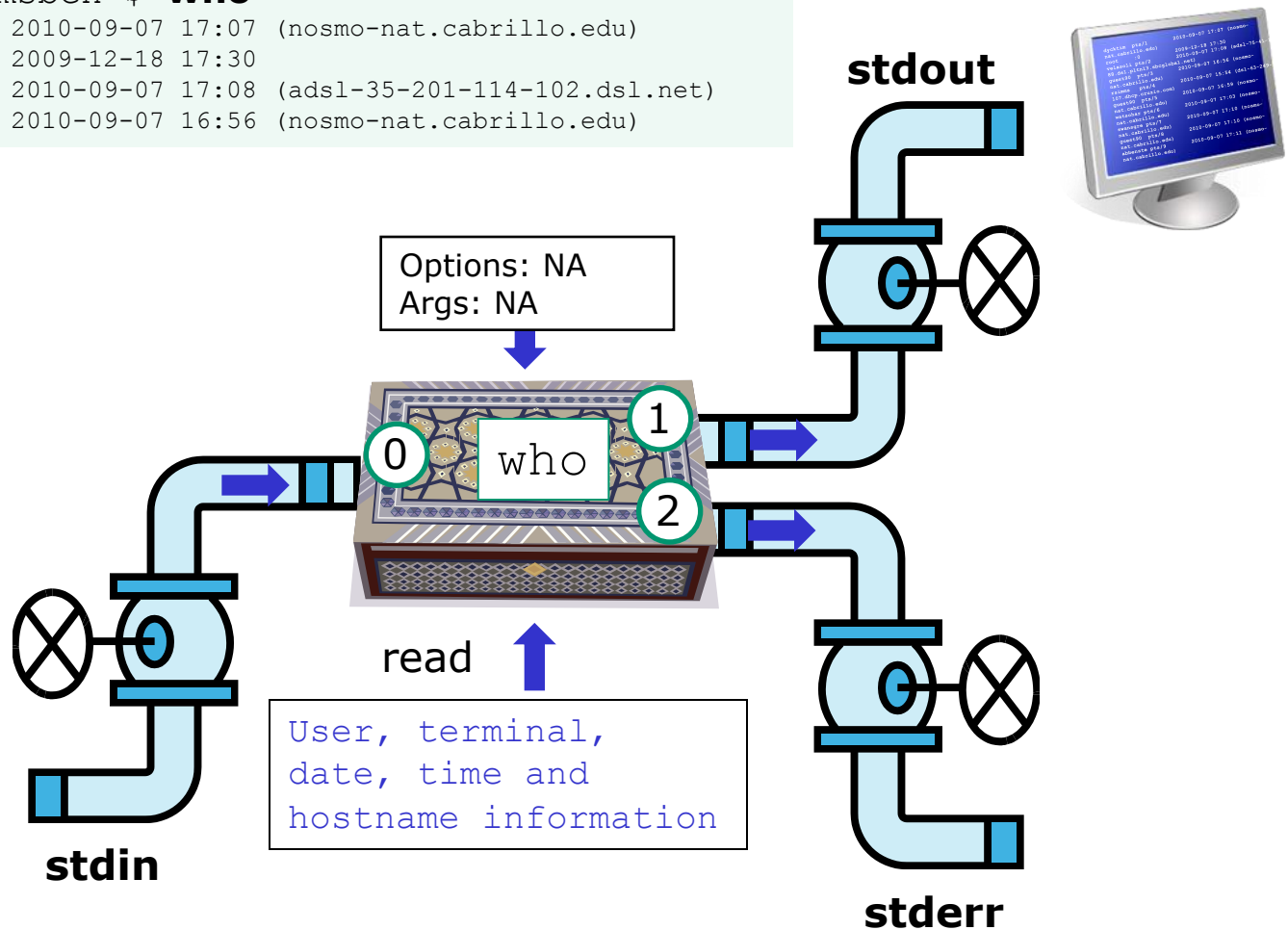
```
/home/cis90/simben $ uname
Linux
```

*The **who** and **uname** commands are examples of commands that get their input from the Operating System*

who command

```
/home/cis90/simmsben $ who
```

```
dycktim pts/1      2010-09-07 17:07 (nosmo-nat.cabrillo.edu)
root      :0        2009-12-18 17:30
velasoli pts/2      2010-09-07 17:08 (adsl-35-201-114-102.dsl.net)
guest90   pts/3      2010-09-07 16:56 (nosmo-nat.cabrillo.edu)
```



The **who** command is an example of a command that gets its input from the Operating System

Class Activity

Where is this **ps** command getting its input from?

```
/home/cis90/simben $ ps
  PID TTY          TIME CMD
 26981 pts/2        00:00:00 bash
 28587 pts/2        00:00:00 ps
/home/cis90/simben $
```

Type your answer in the chat window



Command Syntax

(grammar lesson)



from Dictionary.com

parse [pahrs, pahrz] **verb, parsed, pars-ing.**
verb (used with object)

1. to analyze (a sentence) in terms of grammatical constituents, identifying the parts of speech, syntactic relations, etc.
2. to describe (a word in a sentence) grammatically, identifying the part of speech, inflectional form, syntactic function, etc.
3. Computers . to analyze (a string of characters) in order to associate groups of characters with the syntactic units of the underlying grammar.

One of the things the shell does is parse what is typed by the user. This results in the command line being analyzed to identify the command, the options, the arguments and any redirection.

Command Syntax

Command**Options****Arguments****Redirection**

Command – is the name of an executable program file.

Options – a special type of argument that is used to control how the program operate operates.

Arguments – the objects the command is directed to work upon. Multiple arguments are separated by spaces.

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 Rules

Command**Options****Arguments****Redirection**

Command – usually at the beginning of the line

Options – follow the command, usually starts with a dash, may be combined after a single “-” or separated by spaces. Note that `-iad` is the same as `-i -a -d`

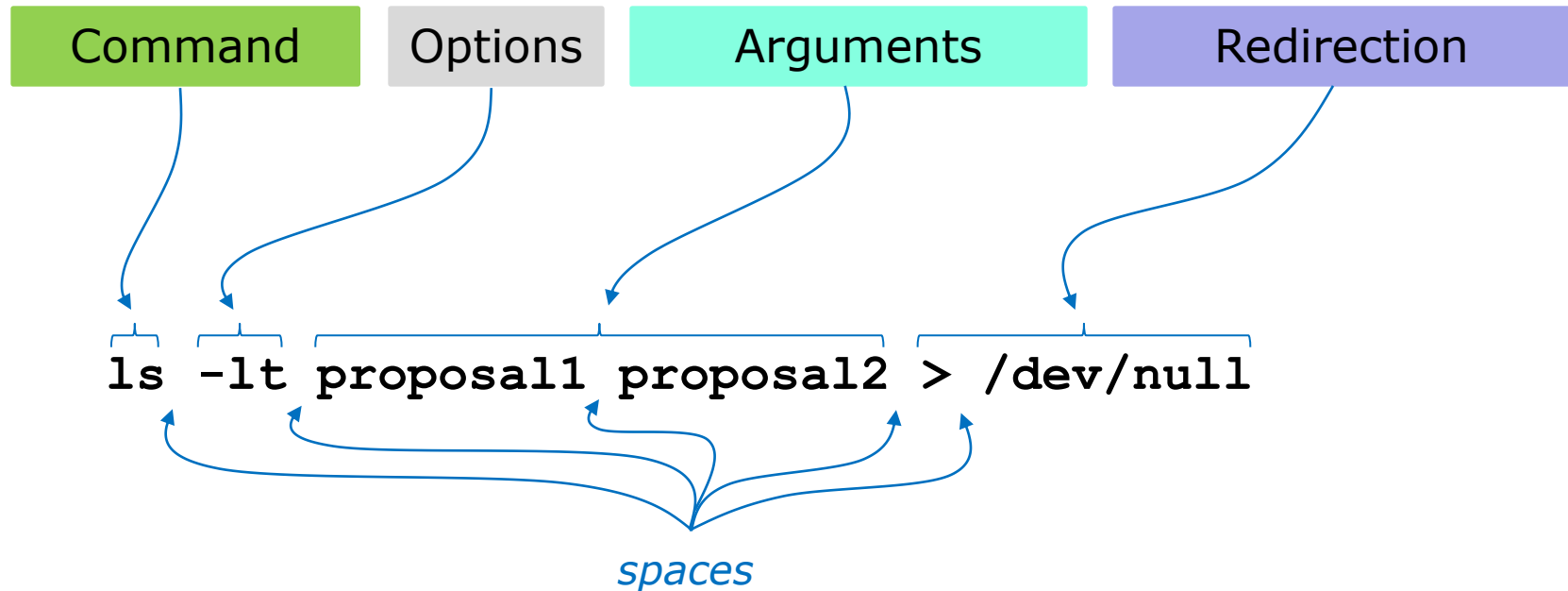
Arguments – follow the options. Multiple arguments must be separated by spaces.

Redirection – Will be a `<`, `>`, `>>`, `2>` or `|` followed by the I/O redirection.

Spaces are required between commands, options, arguments and any redirection

Multiple spaces are treated as a single space (unless inside quotes)

Command Syntax Example



Don't worry now about what the example command above does, for now we just want to be able to parse it into the command, options, arguments and any redirection

More Command Syntax Examples

Command**Options****Arguments****Redirection**

The command syntax is the underlying grammar used to parse the command line

```
/home/cis90/simben $ hostname  
opus.cabrillo.edu
```

```
/home/cis90/simben $ uname -o  
GNU/Linux
```

```
/home/cis90/simben $ ls -ld Poems/  
drwxr-xr-x 5 simben90 cis90 4096 Jan 18 2004 Poems/
```

```
/home/cis90/simben $ ls -li letter > /dev/null
```

More on redirection in later lessons

Parsing Practice

Command Syntax

Command

Options

Arguments

Redirection

```
/home/cis90/simben $ echo I love Linux  
I love Linux
```

Use the chat window to type your answers

Command:

Options:

How many:

What are they:

Arguments:

How many:

What are they:

Redirection:

How many:

What is redirected:

Command Syntax

Command

Options

Arguments

Redirection

```
/home/cis90/simben $ echo I love Linux  
I love Linux
```

Please parse the command line above

Command: echo

Options:

How many: NA
What are they: NA

Arguments:

How many: 3
What are they: I, Love, Linux

Redirection:

How many: NA
What is redirected: NA

Command Syntax

Command

Options

Arguments

Redirection

```
/home/cis90/simben $ ls -ld /bin /usr/bin
drwxr-xr-x 2 root root 4096 Nov 23 13:49 /bin
drwxr-xr-x 2 root root 61440 Nov 23 13:49 /usr/bin
```

Use the chat window to type your answers

Command:

Options:

How many:

What are they:

Arguments:

How many:

What are they:

Redirection:

How many:

What is redirected:

Command Syntax

Command

Options

Arguments

Redirection

```
/home/cis90/simben $ ls -ld /bin /usr/bin  
drwxr-xr-x 2 root root 4096 Nov 23 13:49 /bin  
drwxr-xr-x 2 root root 61440 Nov 23 13:49 /usr/bin
```

Please parse the command line above

Command: ls

Options:

How many: 2
What are they: l, d

Arguments:

How many: 2
What are they: /bin, /usr/bin

Redirection:

How many: NA
What is redirected: NA

Command Syntax

Command

Options

Arguments

Redirection

```
/home/cis90/simben $ ls-ld/bin/usr/bin  
-bash: ls-ld/bin/usr/bin: No such file or directory
```

Use the chat window to type your answers

Command:

Options:

How many:

What are they:

Arguments:

How many:

What are they:

Redirection:

How many:

What is redirected:

Command Syntax

Command

Options

Arguments

Redirection

```
/home/cis90/simben $ ls-ld/bin/usr/bin  
-bash: ls-ld/bin/usr/bin: No such file or directory
```

Please parse the command line above

Command: ls-ld/bin/usr/bin

Options:

How many:	NA
What are they:	NA

Arguments:

How many:	NA
What are they:	NA

Redirection:

How many:	NA
What is redirected:	NA

*Spaces are required between
commands, options,
arguments and any
redirection*

Command Syntax

Command

Options

Arguments

Redirection

```
/home/cis90/simben $ file proposall timecal  
proposall: ASCII English text  
timecal:  shell archive or script for antique kernel text
```

Use the chat window to type your answers

Command:

Options:

How many:

What are they:

Arguments:

How many:

What are they:

Redirection:

How many:

What is redirected:

Command Syntax

Command

Options

Arguments

Redirection

```
/home/cis90/simben $ file proposal1 timecal  
proposal1: ASCII English text  
timecal:  shell archive or script for antique kernel text
```

Please parse the command line above

Command: file

Options:

How many: NA
What are they: NA

Arguments:

How many: 2
What are they: proposal1, timecal

Redirection:

How many: NA
What is redirected: NA

Variables

Shell Variables

- A shell variable gives a name to a location in memory where data can be kept during the session. This data value is lost when a session ends.
- The shell variables used to customize the users environment are called *Environment* variables.
- When parsing, the shell will look for a \$ followed by a variable name and replace it with the value of the variable.

To show the value of a variable use the **echo** command and precede the variable name with a \$

echo \$PS1 *shows the current value of the PS1 variable*

To change the value of a variable, use an = sign with no surrounding blanks and no \$

PS1="Enter next command: " *sets the PS1 prompt variable*

Shell Environment Variables

These variables are automatically set for you when you log in

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 <code>:</code> '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, linux, etc.

Showing environment variable values

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

Shows your terminal type

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

Shows your current working directory

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

Shows your level 1 prompt string

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

Shows your home directory

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

Shows your shell

```
/home/cis90/simben $ 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:.
```

Shows the directories making up your path

Terminal type ≠ Terminal device

The TERM variable holds the terminal type which is different than the terminal device

*Use **tty** to see terminal device*

*Use **echo \$TERM** to see terminal type*

```
simben90@oslab:~  
simben90@oslab.cabrillo.edu's password:  
Last login: Tue Feb  4 18:56:49 2014 from ec2-54-215-232-67.us-west-1.compute.am  
azonaws.com  
  
      ( 'v' )  
    //--\ \  
  ( \ _ _ / )  
    ~ ~ ~ ~  
      Welcome to Opus  
      Serving Cabrillo College  
  
Terminal type? [xterm]  
Terminal type is xterm.  
/home/cis90/simben $ tty  
/dev/pts/1  
/home/cis90/simben $ echo $TERM  
xterm  
/home/cis90/simben $
```

Note the TERM variable gets set every time we log into Opus

Setting Variable Values

To change the value of a variable, use an = sign with no surrounding blanks and no \$

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

*Show the current
terminal type*

```
/home/cis90/simben $ TERM=dumb  
/home/cis90/simben $ echo $TERM  
dumb
```

*Change the terminal
type and display the
new value*

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

*Change the terminal
type back to the
original value*

In Lab 2 you will see what happens when the terminal type is changed

The SHELL variable

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

The SHELL variable will be set to the name of the shell you are running. Benji is running the bash shell.

```
/home/cis90/simben $ ps  
  PID TTY          TIME CMD  
 7364 pts/1        00:00:00 bash  
 7745 pts/1        00:00:00 ps
```

In Lesson 1 we used the ps command to see the shell being run

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

The shell that is run is determined by the entry in /etc/passwd

The PS1 variable

```
/home/cis90/simben $ PS1="By your command > "
```

```
By your command > date
```

```
Mon Sep 3 17:25:32 PDT 2012
```

```
By your command >
```

```
By your command > PS1='What can I do for you $LOGNAME? '
```

```
What can I do for you simben90? date
```

```
Mon Sep 3 17:26:10 PDT 2012
```

```
What can I do for you simben90?
```

```
What can I do for you simben90? PS1='$PWD $ '
```

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

```
Mon Feb 3 18:06:30 PST 2014
```

The PS1 variable determines your shell prompt

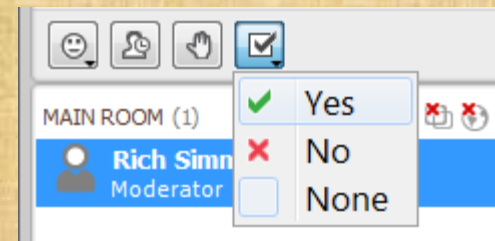
Class Exercise

PS1 "Prompt" variable

Change your prompt to "What is your command master? "

Include a space after the "?"

Give me a green check ✓ if you are successful and a red ✗ if stuck on CCC Confer





*Need a fresh start -- just log out
and back in again and your prompt
will be back to normal!*

Changing the shell prompt

(PS1 variable)

Changing the prompt

There are some special \codes you can insert when setting the prompt

\h gets replaced by the hostname

\W gets replaced by the base working directory

\u gets replaced by the username

```
/home/cis90/simben $ PS1="[\u@\h \W]\$ "
```

```
[simben90@oslab ~]$ date
```

```
Mon Sep 3 17:38:54 PDT 2012
```

```
[simben90@oslab ~]$
```

\\$ gets replaced by a \$ for regular users or # if the root user

user name

hostname

*working directory
(~ is shorthand for the home directory)*

indicates regular user

Changing the prompt

Special Codes	Meaning
\!	history command number
\#	session command number
\d	date
\h	hostname
\n	new line
\s	shell name
\t	time
\u	user name
\w	entire path of working directory
\W	only working directory
\\$	\$ or # (for root user)

The PS1 variable (defines the prompt) can be set to any combination of text, variables and these special codes.

Changing the prompt

Prompt string	Result
PS1='\$PWD \$ '	/home/cis90/simmsben/Poems \$
PS1="\w \$ "	~/Poems \$
PS1="\W \$ "	Poems \$
PS1="\u@\h \$ "	simmsben@opus \$
PS1='\u@\h \$PWD \$ '	simmsben@opus /home/cis90/simmsben/Poems \$
PS1='\u@\h\$HOSTNAME \$PWD \$ '	simmsben@opus.cabrillo.edu /home/cis90/simmsben/Poems \$
PS1='\u \! \$PWD \$ '	simmsben 825 /home/cis90/simmsben/Poems \$
PS1="[\u@\h \W] \$ "	[simmsben@opus Poems] \$

Important: Use single quotes around variables that change. For example if you use \$PWD with double quotes, the prompt will not changes as you change directories! More on this later ...



*Need a fresh start -- just log out
and back in again and your prompt
will be back to normal!*

Listing all the variables

Shell Variables set command

```
/home/cis90/simben $ set
BASH=/bin/bash
BASHOPTS=checkwinsize:cmdhist:expand_aliases:extquote:force_ignores
complete:interactive_comments:login_shell:progcomp:promptvars:sourcepath
BASH_ALIASES=()
BASH_ARGC=()
BASH_ARGV=()
BASH_CMDS=()
BASH_ENV=/home/cis90/simben/.bashrc
BASH_LINENO=()
BASH_SOURCE=()
BASH_VERSINFO=([0]="4" [1]="1" [2]="2" [3]="1" [4]="release" [5]="i386-
redhat-linux-gnu")
BASH_VERSION="4.1.2(1)-release"
COLORS=/etc/DIR_COLORS
COLUMNS=123
CVS_RSH=ssh
DIRSTACK=()
EUID=1001
GROUPS=()
G_BROKEN_FILENAMES=1
HISTCONTROL=ignoredups
HISTFILE=/home/cis90/simben/.bash_history
HISTFILESIZE=1000
HISTSIZE=1000
HOME=/home/cis90/simben
HOSTNAME=oslab.cabrillo.edu
HOSTTYPE=i386
ID=1001
IFS=$' \t\n'
IGNOREEOF=10
LANG=en_US.UTF-8
LESSOPEN='|/usr/bin/lesspipe.sh %s'
LINES=38
LOGNAME=simben90
```

*The **set** command shows all shell variables including the special environment variables.*

```
LS_COLORS='rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;3
0;01:cd=40;33;01:or=40;31;01:mi=01;05;37;41:su=37;41:sg=30;43:ca=30;41:tw=
30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz
=01;31:*.lzh=01;31:*.lзма=01;31:*.tlz=01;31:*.txz=01;31:*.zip=01;31:*.z=01
;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lz=01;31:*.xz=01;31:*.bz2=01;31:*.tb
z=01;31:*.tbz2=01;31:*.bz=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=0
1;31:*.rar=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;3
1:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35
:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:
*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.
mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.ogm=01;35:*.mp4=01;35:*.
m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.as
f=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=
01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;3
5:*.emf=01;35:*.axv=01;35:*.anx=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=01;36:
*.au=01;36:*.flac=01;36:*.mid=01;36:*.midi=01;36:*.mka=01;36:*.mp3=01;36:*.
mpc=01;36:*.ogg=01;36:*.ra=01;36:*.wav=01;36:*.axa=01;36:*.oga=01;36:*.sp
x=01;36:*.xspf=01;36:'
MACHTYPE=i386-redhat-linux-gnu
MAIL=/var/spool/mail/simben90
MAILCHECK=60
OLDPWD=/bin
OPTERR=1
OPTIND=1
OSTYPE=linux-gnu
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:.
PIPESTATUS=([0]="127")
PPID=17309
PROMPT_COMMAND='printf "\033]0;%s@%s:%s\007" "${USER}" "${HOSTNAME%%.*}"
"${PWD}/${HOME}/~}"'
PS1='$PWD $ '
PS2='> '
PS4='+ '
PWD=/home/cis90/simben
QTDIR=/usr/lib/qt-3.3
QTINC=/usr/lib/qt-3.3/include
QTLIB=/usr/lib/qt-3.3/lib
SELINUX_LEVEL_REQUESTED=
SELINUX_ROLE_REQUESTED=
SELINUX_USE_CURRENT_RANGE=
SHELL=/bin/bash
SHELLOPTS=braceexpand:emacs:hashall:histexpand:history:ignoreeof:interacti
ve-comments:monitor
SHLVL=1
SSH_CLIENT='50.0.68.235 51849 2220'
SSH_CONNECTION='50.0.68.235 51849 172.30.5.20 2220'
SSH_TTY=/dev/pts/2
TERM=xterm
UID=1001
USER=simben90
USERNAME=
_=ser
colors=/etc/DIR_COLORS
/home/cis90/simben $
```

Shell (Environment) Variables

env command

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

```
HOSTNAME=oslabb.cabrillo.edu
```

```
SELINUX_ROLE_REQUESTED=
```

```
TERM=xterm
```

```
SHELL=/bin/bash
```

```
HISTSIZE=1000
```

```
SSH_CLIENT=50.0.68.235 51849 2220
```

```
SELINUX_USE_CURRENT_RANGE=
```

```
QTDIR=/usr/lib/qt-3.3
```

```
QTINC=/usr/lib/qt-3.3/include
```

```
SSH_TTY=/dev/pts/2
```

```
USER=simben90
```

```
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=01;05;37;41:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz=01;31:*.lzh=01;31:*.lzm=01;31:*.tlz=01;31:*.txz=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lz=01;31:*.xz=01;31:*.bz2=01;31:*.tbz=01;31:*.tbz2=01;31:*.bz=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.rar=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.axv=01;35:*.anx=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=01;36:*.au=01;36:*.flac=01;36:*.mid=01;36:*.midi=01;36:*.mka=01;36:*.mp3=01;36:*.mpc=01;36:*.ogg=01;36:*.ra=01;36:*.wav=01;36:*.axa=01;36:*.oga=01;36:*.spx=01;36:*.xspf=01;36:
```

```
USERNAME=
```

```
MAIL=/var/spool/mail/simben90
```

```
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:.
```

```
PWD=/home/cis90/simben
```

```
LANG=en_US.UTF-8
```

```
SELINUX_LEVEL_REQUESTED=
```

```
HISTCONTROL=ignoredups
```

```
SHLVL=1
```

```
HOME=/home/cis90/simben
```

```
BASH_ENV=/home/cis90/simben/.bashrc
```

```
LOGNAME=simben90
```

```
QTLIB=/usr/lib/qt-3.3/lib
```

```
CVS_RSH=ssh
```

```
SSH_CONNECTION=50.0.68.235 51849 172.30.5.20 2220
```

```
LESSOPEN=|/usr/bin/lesspipe.sh %s
```

```
G_BROKEN_FILENAMES=1
```

```
_=/bin/env
```

```
OLDPWD=/bin
```

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

*The **env** command shows just the environment variables (a subset of the shell variables)*

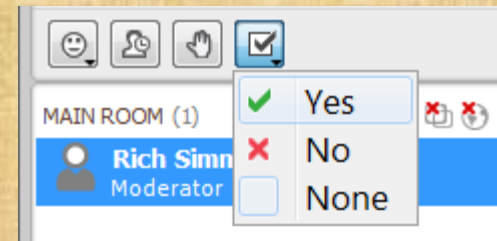
Class Exercise

PS1 "Prompt" variable

Change your prompt to "What is your command master? "

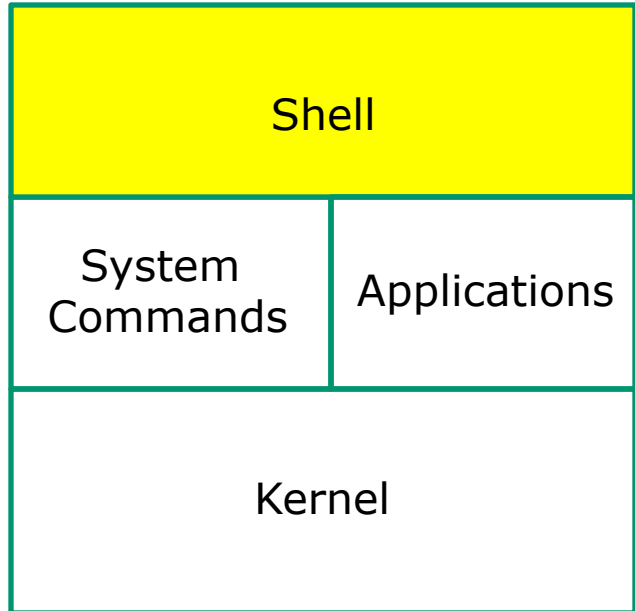
Include a space after the ?

Give me a green check ✓ if you are successful and a red ✗ if stuck on CCC Confer



The Shell (Deep Dive)

The Shell

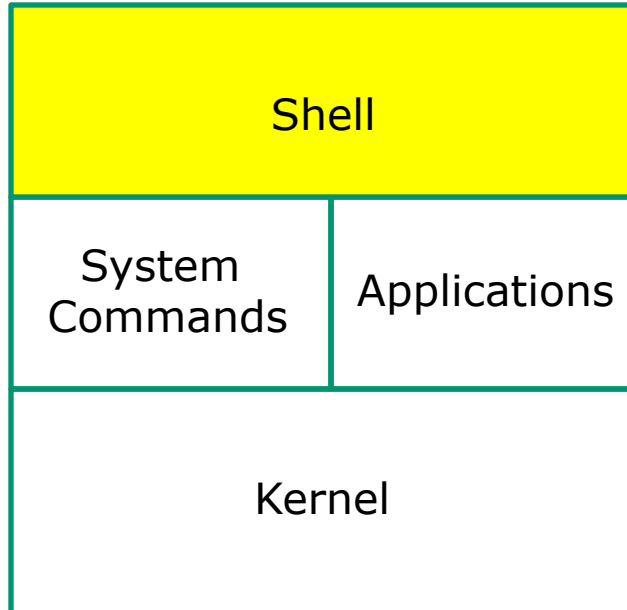


- Allows users to interact with the computer via a “**command line**”.
- **Prompts** for a command, parses the command, finds the right program and gets that program executed.
- Is called a “**shell**” because it hides the underlying operating system.
- Multiple shell programs are available: **sh** (Bourne shell), **bash** (Bourne Again shell), **csh** (C shell), **ksh** (Korn shell).
- The shell is a **user interface** and a **programming language** (scripts).
- GNOME and KDE desktops could be called **graphical shells**





Life of the Shell



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





Life of the Shell

Example:

```
/home/cis90/simben $ ls -lt proposal1 proposal2
-rw-r--r--. 1 simben90 cis90 1074 Aug 26 2003 proposal1
-rw-r--r--. 1 simben90 cis90 2175 Jul 20 2001 proposal2
/home/cis90/simben $
```

Shell Steps

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

Lets take a deep dive into how a command gets executed.

Note it is always a team effort by both the shell and the command.



Life of the Shell

Shell Steps

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

1) Prompt user for a command

Example:

*The shell begins by outputting the prompt
(which is based on the PS1 variable)*

```
/home/cis90/simben $ ls -lt proposal1 proposal2
```

Then you type the command

FYI, you can mimic outputting the prompt yourself with these commands:

```
/home/cis90/simben $ echo $PS1 to show value of PS1 variable
```

```
$PWD $
```

```
/home/cis90/simben $ echo $PWD $ echo the output of the  
previous command
```

```
/home/cis90/simben $ was output by the echo command above
```

```
/home/cis90/simben $ was output by the shell (the same output)
```



Life of the Shell

2) Parse command user typed

Shell Steps

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

Example:

```
ls -lt proposal1 proposal2
```

During the parse step the shell identifies all options & arguments, handles any metacharacters and redirection

- Command = ls
- 2 Options = l, t
- 2 Arguments = proposal1, proposal2
- 1 Redirection = NA



Life of the Shell

3) Search path for the program to run

Shell Steps

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

ls -lt proposal1 proposal2

Use this command to see the path directories (separated by ':'s) on your path

```
/home/cis90/simben $ 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 shell will search each directory in order for an **ls** command*

```
/usr/lib/qt-3.3/bin no ls command found here
/usr/local/bin no ls command found here
/bin YES! - an ls command is in the /bin directory
/usr/bin
/usr/local/sbin
/usr/sbin
/sbin
/home/cis90/simben/../../bin
/home/cis90/simben/bin
.
```

Note: If the shell cannot find the command on the path it will output "command not found"

Try mimicking what the shell does to search for ls:

```
/home/cis90/simben $ ls /usr/lib/qt-3.3/bin/ls
ls: cannot access /usr/lib/qt-3.3/bin/ls: No such file or directory
```

```
/home/cis90/simben $ ls /usr/local/bin/ls
ls: cannot access /usr/local/bin/ls: No such file or directory
```

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



Life of the Shell

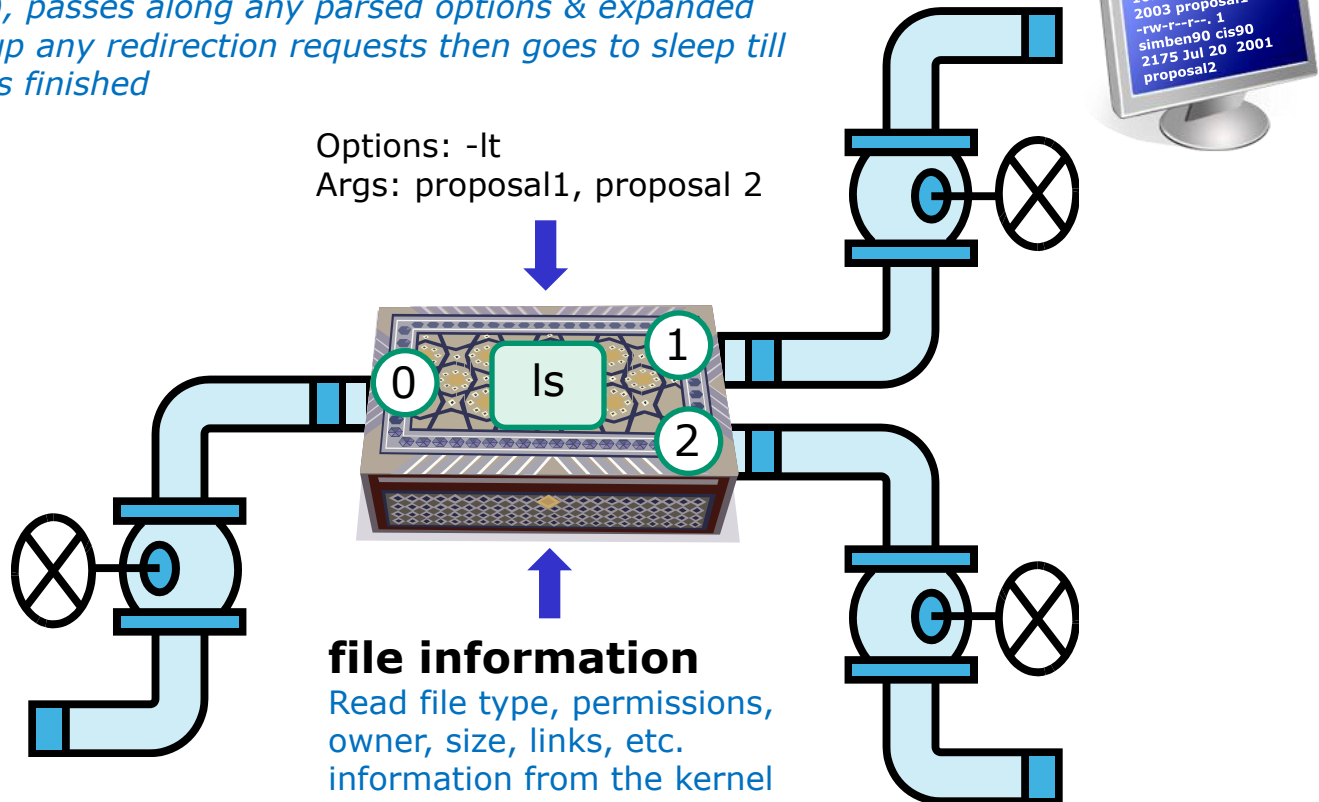
Shell Steps

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

4) Execute the command

```
ls -lt proposal1 proposal2
```

Invokes the kernel to load the program into memory (which becomes a process), passes along any parsed options & expanded arguments, hooks up any redirection requests then goes to sleep till the new process has finished





Life of the Shell

5) Nap while the command (process) runs to completion

(The shell, itself a loaded process, goes into the sleep state and waits till the command process is finished)

Shell Steps

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

```
/home/cis90/simben $ ls -lt proposal1 proposal2
-rw-r--r--. 1 simben90 cis90 1074 Aug 26 2003 proposal1
-rw-r--r--. 1 simben90 cis90 2175 Jul 20 2001 proposal2
```

The shell sleeps while the ls process outputs these two lines



Life of the Shell

6) And do it all over again
... go to step 1

Shell Steps

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



Life of the Shell

A /home/cis90/simben \$ **Ls -lt proposal1 proposal2**

-bash: Ls: command not found

What's wrong?

Who output the error?

B /home/cis90/simben \$ **ls -lt proposal1 proposal5**

ls: cannot access proposal5: No such file or directory
-rw-r--r--. 1 simben90 cis90 1074 Aug 26 2003 proposal1

What's wrong?

Who output the error?

C /home/cis90/simben \$ **ls -lw proposal1 proposal2**

ls: invalid line width: proposal1

What's wrong?

Who output the error?

D /home/cis90/simben \$ **ls -lt proposal1proposal2**

ls: cannot access proposal1proposal2: No such file or directory

What's wrong?

Who output the error?

E /home/cis90/simben \$ **ls-lt proposal1 proposal2**

-bash: ls-lt: command not found

What's wrong?

Who output the error?

Meta- characters

Metacharacters

When parsing, the shell gives special meaning to metacharacters

" - use double quotes to preserve blanks and allow variable expansion

' - use single quotes to preserve blanks and block variable expansion

\$ - use to show the value rather than the name of a variable

;- allows multiple commands on one line

<enter key> - The invisible newline control character marking the end of a command

= - use to set variables to new values

\ - removes (escapes) the special powers of a metacharacter

Other metacharacters we will learn about later include:

*?, *, <, >, >>, !, |, [], {}, &, && and ||*

Metacharacters - quotes

- Double " quotes allow variable expansion
- Single ' quotes block variable expansion
- Both double and single quotes preserve blanks

```
/home/cis90/simben $ echo I am $LOGNAME (3 arguments)
I am simben90 Extra blanks ignored, variable expanded
```

```
/home/cis90/simben $ echo "I am $LOGNAME" (1 argument)
I am simben90 Extra blanks preserved, variable expanded to show value
```

```
/home/cis90/simben $ echo 'I am $LOGNAME' (1 argument)
I am $LOGNAME Extra blanks preserved, variable expansion blocked
```

Double quotes called weak quotes because they allow the shell to expand variables. Single quotes are called strong quotes because they block the shell from expanding variables.

Metacharacters - quotes

```
/home/cis90/simben $ echo '"double quotes"'  
"double quotes"
```

```
/home/cis90/simben $ echo "'single quotes'"  
'single quotes'
```

Tip: single quotes can be used to output double quotes and vice-versa

Metacharacters - <enter key>

<enter key> - The invisible *newline* control character marking the end of a command



```
[rsimms@opus ~]$ ps
  PID TTY          TIME CMD
 19015 pts/0    00:00:00 bash
 19378 pts/0    00:00:00 ps

[rsimms@opus ~]$ hostname
opus.cabrillo.edu

[rsimms@opus ~]$ echo "Use <enter key> to end the command"
Use <enter key> to end the command
```

Pressing the Enter key here generates an invisible <newline> character

Metacharacters - \ (backslash)

The back slash \ removes (escapes) the special powers of a metacharacter

```
[rsimms@oslab ~]$ echo a b c d e f
a b c d e f
```

```
[rsimms@opus ~]$ echo a b c \
> d e f
a b c d e f
```

*Escape the invisible newline <enter key>
which marks the end of a command*

```
[rsimms@opus ~]$ echo $PS1
[\u@\h \W]\$
```

```
[rsimms@opus ~]$ echo \$PS1
$PS1
```

*Escape the \$ (which shows
the value of the variable)*

```
[rsimms@opus ~]$ echo "Hello World"
Hello World
```

```
[rsimms@opus ~]$ echo \"Hello World\"
\"Hello World\"
```

*Escape the double quote
marks*

Metacharacters - ; (semi-colon)

The semi-colon ; allows multiple commands on one line

```
[simmsben@opus Poems]$ hostname; uname; echo $LOGNAME; ls  
opus.cabrillo.edu  
Linux  
simmsben  
ant  Blake  nursery  Shakespeare  twister  Yeats
```

*Four commands on
one line*

Shortcuts

More on the Command Line

Handy Shortcuts

- Use up and down arrows to “retype” previous commands
- Left and right arrow for editing current command
- Use <tab> to complete filenames automatically

```

/home/cis90/simben $ hostname; name; echo $LOGNAME; ls Poems/Blake/
oslab.cishawks.net
- bash: name: command not found
simben90
jerusalem tiger
/home/cis90/simben $ hostname; uname; echo $LOGNAME; ls Poems/Blake/
oslab.cishawks.net
Linux
simben90
jerusalem tiger
  
```

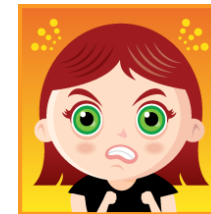
Press <tab> after the P and B and the shell fills in the rest

Press up arrow and the shell retypes the previous command

Use the left arrow to backup and fix the typo (uname instead of name)

Life without a path

-bash: xxxx: command not found



Don't get mad, just fix your path!

The Path

The shell uses your path to locate commands to execute

- A path is a ordered set of directories along which the shell will search to locate commands to execute
- The path is defined by the PATH variable
- Show your path with **echo \$PATH**
- If you specify a command xxxx that the shell cannot find on the path it will print the following error message:

-bash: xxxx: command not found
- To run a command that is not on your path the complete absolute pathname must be specified. e.g. /usr/bin/uname

The Path

Use this command to see the directories (separated by :’s) on your path

```
/home/cis90/simben $ 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 shell will search for the ls command along the path in this order:

```
/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
```

```
.
```



yes, . is a directory too and it is whatever directory you have currently changed into

Experiment – Breaking the Path

The **echo**
command is
built into bash

```
/home/cis90/simben $ type echo ps tty
echo is a shell builtin
ps is /bin/ps
tty is /usr/bin/tty
```

the **ps**
command is in
the **/bin**
directory

The **tty** command
is in the **/usr/bin**
directory



/usr/bin



/bin

Experiment – Breaking the Path

*Default
path*

```
/home/cis90/simben $ echo I love Linux
I love Linux
/home/cis90/simben $ date
Mon Sep  3 15:17:52 PDT 2012
/home/cis90/simben $ tty
/dev/pts/2
/home/cis90/simben $
```

TROUBLE!

```
/home/cis90/simben $ PATH=""
/home/cis90/simben $ echo $PATH

/home/cis90/simben $
```

*Break the path by
setting it to null*

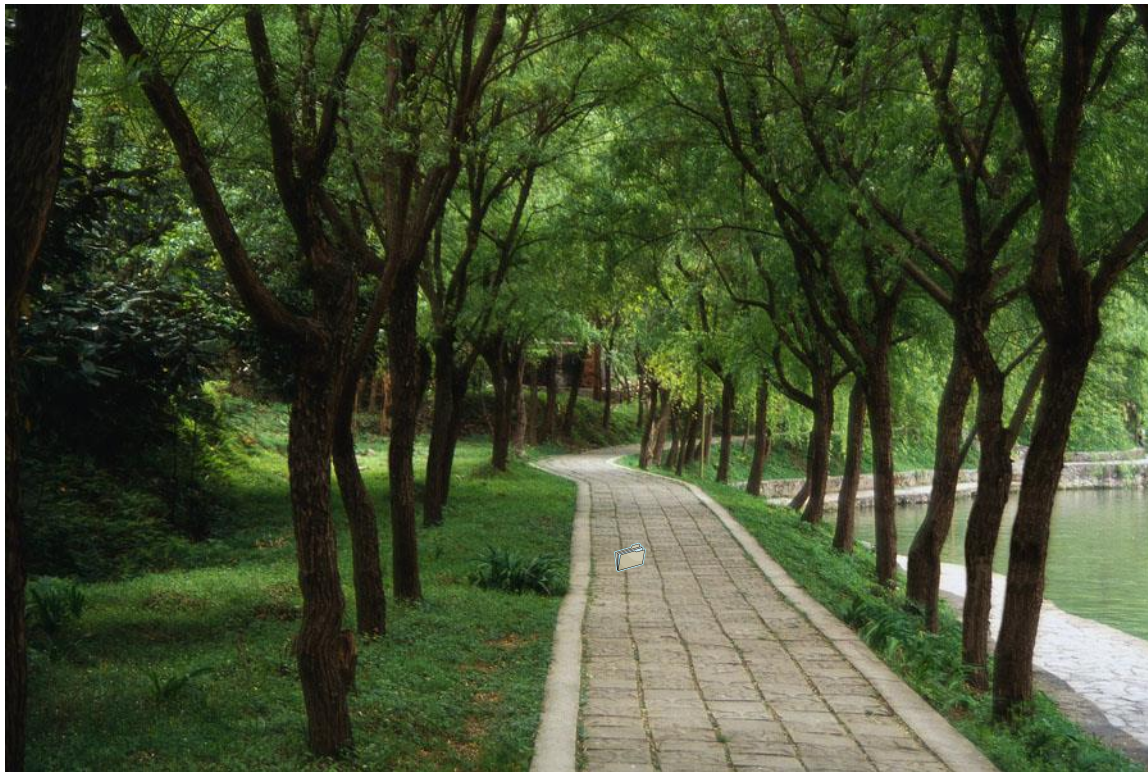
No path

```
/home/cis90/simben $ echo I love Linux
I love Linux
/home/cis90/simben $ date
-bash: date: No such file or directory
/home/cis90/simben $ tty
-bash: tty: No such file or directory
```

*Only **echo**
works because
it is built into
the shell!*

```
/home/cis90/simben $ echo $PATH
```

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



There is nothing on the path!

Experiment – Restoring the Path

```
/home/cis90/simben $ PATH=/bin
/home/cis90/simben $ echo $PATH
/bin
/home/cis90/simben $
```

*Add the /bin
directory to the path*

*date works
because it
resides in the
/bin directory
which is now
on the path*

```
/home/cis90/simben $ echo I love Linux
I love Linux
/home/cis90/simben $ date
Mon Sep  3 15:24:19 PDT 2012
/home/cis90/simben $ tty
-bash: tty: No such file or directory
```

*echo works
because it is built
into the shell*

*tty does not work because it is
in the /usr/bin directory which is
not on the path*

```
/home/cis90/simben $ echo $PATH  
/bin  
/home/cis90/simben $
```



Experiment – Restoring the Path

```
/home/cis90/simben $ PATH=$PATH:/usr/bin  
/home/cis90/simben $ echo $PATH  
/bin:/usr/bin  
/home/cis90/simben $
```

*Append the
/usr/bin directory
to the path*

```
/home/cis90/simben $ echo I love Linux  
I love Linux  
/home/cis90/simben $ date  
Mon Sep  3 15:24:19 PDT 2012  
/home/cis90/simben $ tty  
/dev/pts/2
```

All three commands work because /bin and /usr/bin are on the path.

The shell will only run commands found in the directories that make up the path

```
/home/cis90/simben $ echo $PATH  
/bin:/usr/bin  
/home/cis90/simben $
```





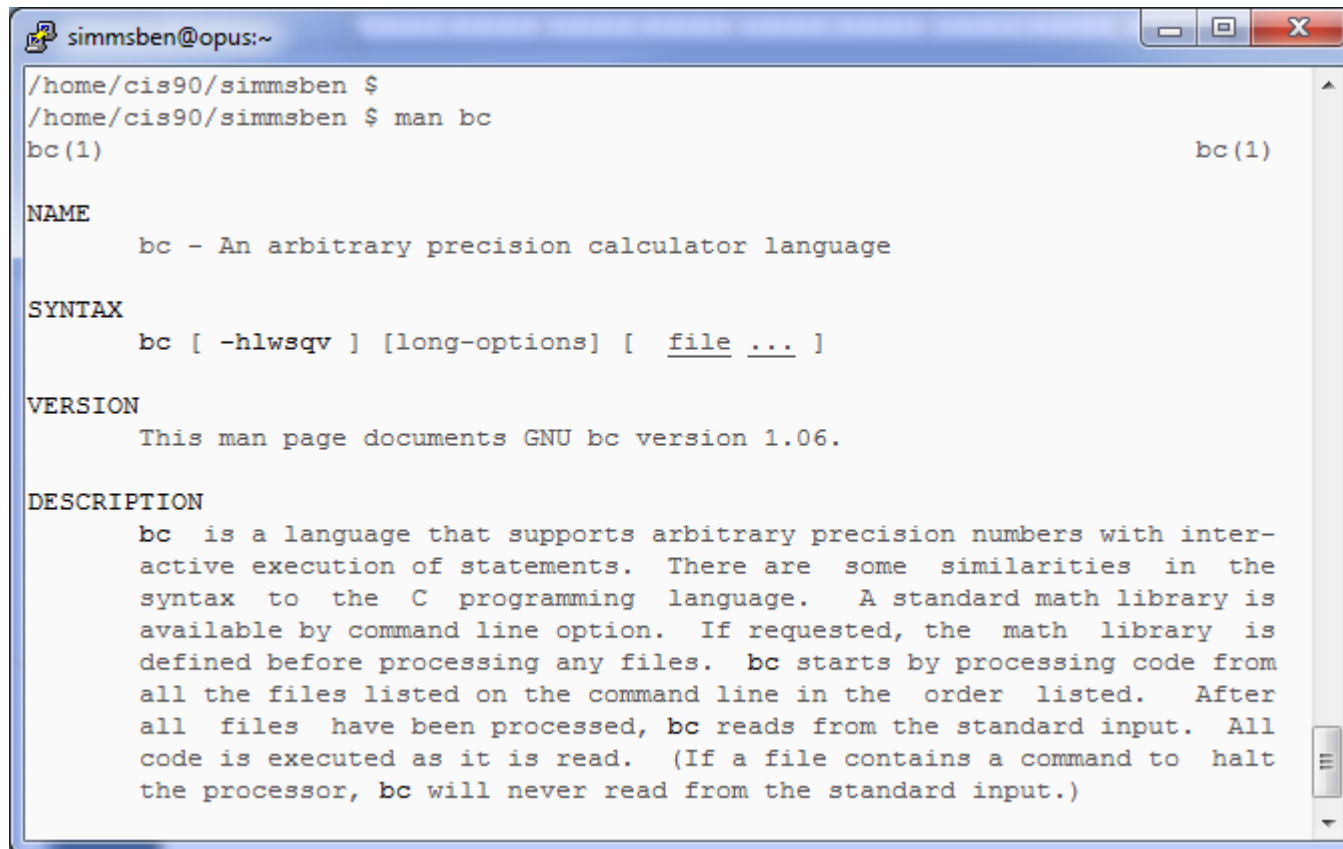
*Need a fresh start -- just log out
and back in again and your path
will be back to normal!*

Docs

Using man (manual) pages

Type the **man** command followed by the name of the command you want documentation on.

Example: **man bc**



```

simmsben@opus:~
/home/cis90/simmsben $
/home/cis90/simmsben $ man bc
bc(1)                                     bc(1)

NAME
    bc - An arbitrary precision calculator language

SYNTAX
    bc [ -hlwsqv ] [long-options] [ file ... ]

VERSION
    This man page documents GNU bc version 1.06.

DESCRIPTION
    bc is a language that supports arbitrary precision numbers with inter-
    active execution of statements. There are some similarities in the
    syntax to the C programming language. A standard math library is
    available by command line option. If requested, the math library is
    defined before processing any files. bc starts by processing code from
    all the files listed on the command line in the order listed. After
    all files have been processed, bc reads from the standard input. All
    code is executed as it is read. (If a file contains a command to halt
    the processor, bc will never read from the standard input.)
  
```



Use these
keys to scroll



Use q key to
quit

Using Google

Do a Google search on "linux xxx command" where xxx is the command you want documentation for.

Example: **google** linux bc command

The image shows two overlapping browser windows. The left window displays a Google search for "linux bc command", showing approximately 1,180,000 results. The right window shows the "bc - Linux Command" page from about.com. This page includes a PayPal advertisement at the top, followed by the "About.com: Linux" header. The main content area is titled "Linux / Unix Command: bc" and lists the command's name, syntax, and description. The description states that bc is an arbitrary precision calculator language that supports arbitrary precision numbers and has a syntax similar to C. It also mentions that a standard math library is available via the "l" command-line option.

Google Search Results (Left Window):

- Search query: linux bc command
- Results: About 1,180,000 results (0.24 seconds)
- Top results include:
 - bc - Linux Command - Unix
 - Linux and UNIX bc command
 - command-line calculations u
 - Linux bc Command - Basic
 - bc: A Handy Utility | Linux Jo

about.com: Linux (Right Window):

Search: linux bc command

Filed In: Linux

Linux / Unix Command: bc

Command Library

NAME

bc - An arbitrary precision calculator language

SYNTAX

bc [-hlwsvq] [long-options] [file ...]

DESCRIPTION

bc is a language that supports arbitrary precision numbers with interactive execution of statements. There are some similarities in the syntax to the C programming language. A standard math library is available by command line option. If requested, the math library is defined before processing any files. bc starts by processing code from all the files listed

Advertisements:

- PayPal merchants are virtually immune to identity theft and scammers.
- PayPal MERCHANT SERVICES
- Free Linux Newsletter! (Sign up)
- PayPal MERCHANT SERVICES
- Walmart
- lenovo.
- eToys.com

Other Documentation

- **whatis** *command* *same as the **man -f** command*
- **apropos** *command* *same as the **man -k** command*
- **info** *command*

Documentation

Two of my favorite documentation links

Rich's Cabrillo College CIS Classes Resources

Home **Resources** Forums CIS Lab CTC

Login
Flashcards
Admin

CIS 90
Previous Classes

103 days till term ends!

Cabrillo College
Web Advisor
CCC Confer
Static IPs
Quick Ref
VM Repairs
GAH!

Links

Instructors

- Linux Master Jim
- Programming Master Ed
- Network Master Gerlinde
- Network Master Rick
- Web Master John
- Windows Master Gary

Clubs

- GNU Linux Users Group

Departments

- CNSA
- CIS
- CS

Crib Sheets

- Ollie Wright (CIS 90)

Documentation

- TLDP
- LINFO

Animations

- Linux network technologies

Getting Linux

- Linux ISOs
- Kernels
- RPMs (rpmfind)
- RPMs (pbone)

Tools and Software

- Apache
- Bastille
- cygwin
- DOS boot disks
- Dyn
- John
- MS
- All
- Qu
- su
- Tri
- Vir
- VM
- Wi

Howtos

- HowtoForge
- email
- DNS
- Ethernet (NIC drivers)
- NFS
- NIS
- PPP
- Putty SSH Keys
- sed

Google | 0 un... | Yahoo... | Rich's... | The Li... | Cabril... | linux I... |

http://tldp.org/

The Linux Documentation Project

2010-09-06

LDP Worldwide

- Mirrors
- Non-English info
- Translation effort
- Translated Guides
- Translated HOWTOs
- Printed books
- Main site

LDP Information

- FAQ
- Manifesto / license
- History
- Volunteers/Staff
- Job Descriptions
- Mailing lists
- LDP Weekly News
- Archives / RSS feed
- IRC
- Feedback
- Apparel

Workshop

LDP Wiki: The LDP Wiki is the entry point for any work in progress
Members | Authors | Visitors

Documents

HOWTOs: subject-specific help
latest updates | main index | browse by category

Guides: longer, in-depth books
latest updates / main index

FAQs: Frequently Asked Questions
latest updates / main index

man pages: help on individual commands (20060810)

Search / Resources

Links
OMF search

Google | 0 un... | Yahoo... | Rich's... | The Li... | Cabril... | linux I... |

http://www.linfo.org/index.html

The Linux Information Project

Welcome to The Linux Information Project (LINFO). This project is dedicated to providing high quality, comprehensive and easily accessible information about **Linux** and other **free software**. (New to Linux? Start [here](#).)

New on This Site:

- October 27: [root Definition](#) page updated.
- October 19: [Hard Link Definition](#) page added.
- October 12: [Characters: A Brief Introduction](#) page updated.
- October 03: [Byte Definition](#) page updated.
- September 27: [PDP-7 Definition](#) page updated.
- September 24: [The mount Command](#) page added.
- September 20: [The head Command](#) page updated.

Site Contents:

The Linux Documentation and Information Projects

Wrap up

Lab 2 - Using Commands

Cabrillo College

CIS 90 Linux Lab Exercise

Lab 2: Using Commands
Fall 2014

Lab 2: Using Commands

The purpose of this lab is to explore command usage with the shell and miscellaneous UNIX commands.

Preparation

Everything you need to do this lab can be found in the Lesson 2 materials on the CIS 90 Calendar: <http://simms-teach.com/cis90calendar.php>. Review carefully all Lesson 2 slides, even those that may not have been covered in class.

Check the forum at: <http://oslab.cis.cabrillo.edu/forum/> for any tips and updates related to this lab. The forum is also a good place to ask questions if you get stuck or help others.

If you would like some additional assistance come to the CIS Lab on campus where you can get help from instructors and student lab assistants: <http://webhawks.org/~cislab/>.

Procedure

This lab must be done on Opus to get credit

Please log into the Opus server using your personal account. You will need to use the following commands in this lab.

banner	clear	finger	man	uname
bash	date	history	passwd	whatis
bc	echo	id	ps	who
cal	exit	info	type	

Only your command history along with the three answers asked for by the submit script will be graded. You must issue each command below (exactly). Rather than submitting answers to any questions asked below you must instead issue the correct commands to answer them. Your command history will be scanned to verify each step was completed.

- This lab **MUST** be done on Opus to get credit
- You don't need to turn in answers for steps 1-22. However I will check your command history to verify you entered the correct commands to answer those questions.
- There are three questions to answer on the **submit** script.

New commands:

apropos	- search for string in whatis database
bc	- binary calculator
cat	- print file(s)
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

Next Class

Assignment: Check Calendar Page on web site to see what is due next week.

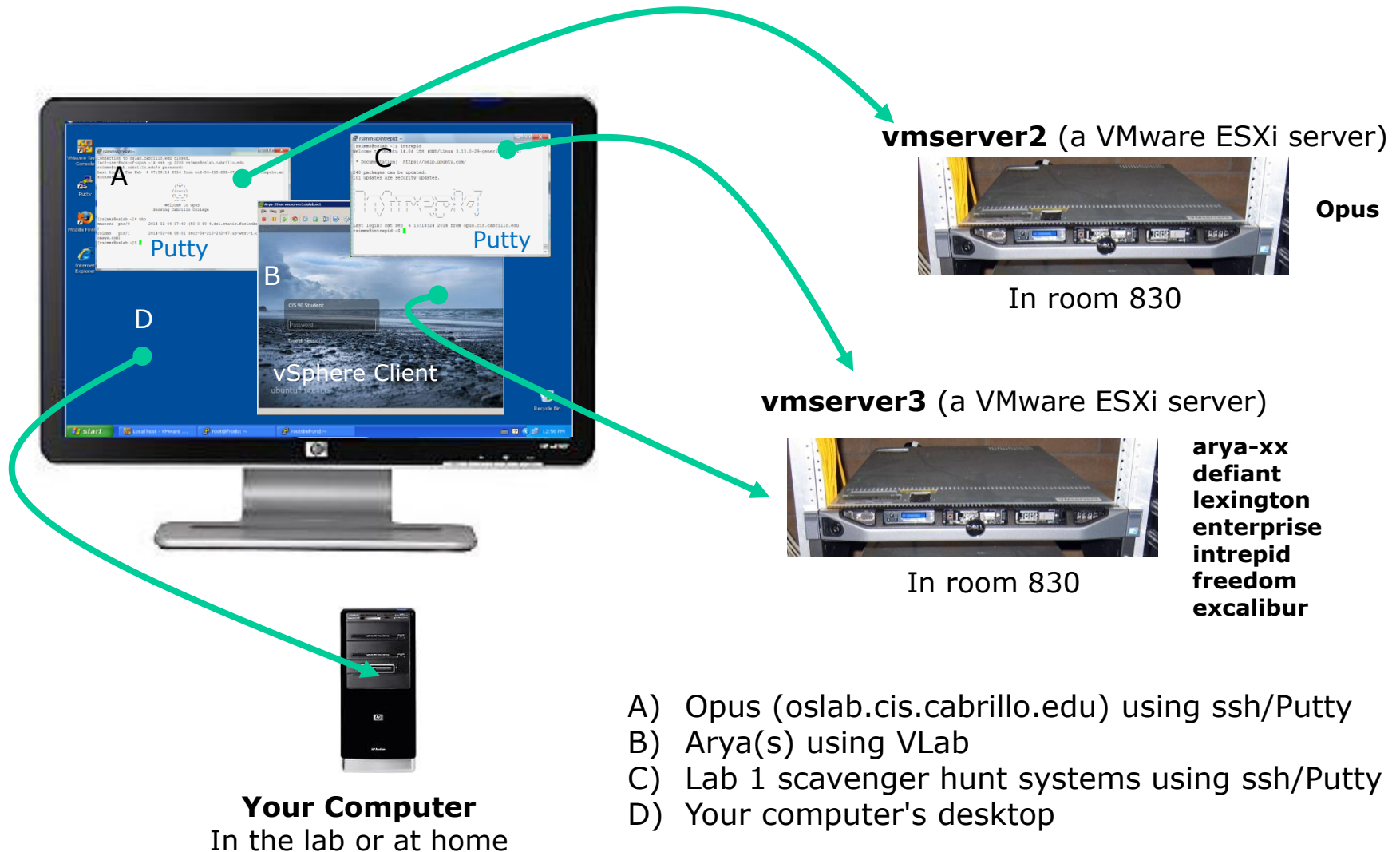
Lab #2

Quiz questions for next class:

- Which four directories typically contain the majority of the UNIX/Linux system commands?
- How do you show your path?
- What command would allow you to view the manual page for the who command?

Backup

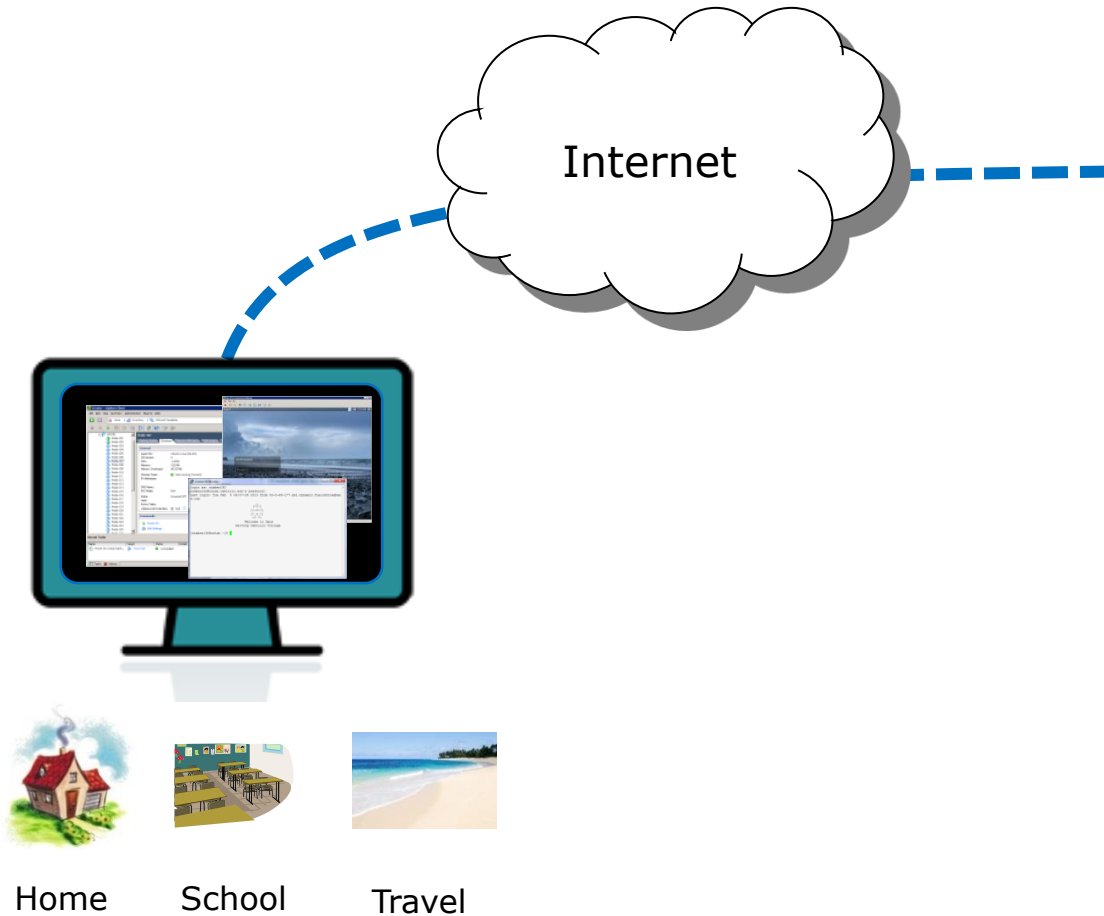
Logging into the various CIS 90 systems from home or the lab



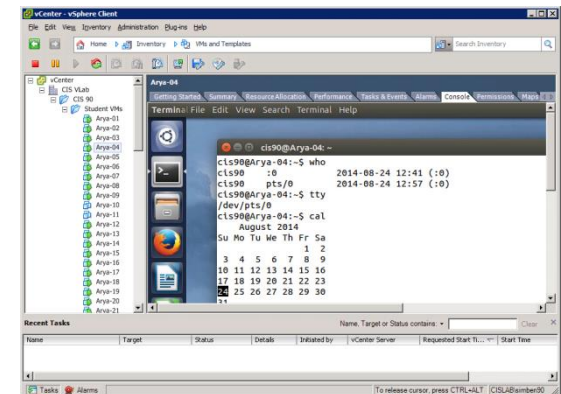
Using CIS VLab (Virtual Lab)

Third driving lesson

Accessing CIS VLab VMs



CIS Lab servers on
the Aptos campus





Rich's Cabrillo College CIS Classes Home Page

[Home](#)
[Resources](#)
[Forums](#)
[CIS Lab](#)
[Blackboard](#)

[Login](#)
[Flashcards](#)
[Admin](#)

[CIS 90](#)
[Previous Classes](#)

0 days till term starts!

[Cabrillo College](#)
[Web Advisor](#)
[Commands and Files](#)

[VLab RDP file](#)

[CIS 90 VLab VM Assignments](#)

[RIP Dennis Ritchie](#)

Rich Simms



Contact

- Email: risimms@cabrillo.edu
- Office hours: [directory page](#)

Fall 2013 Cabrillo Linux Classes

- Introduction to UNIX/Linux (CIS 90) - Rich teaching
- UNIX/Linux System Administration (CIS 191AB) - [Michael Matara](#) teaching

[Metal](#)
[Sitemap](#)


[Credits](#)
[Earth](#)

CIS 90 VLab Assignments

Assignment	Due Date
Assignment 1	Week 1
Assignment 2	Week 2
Assignment 3	Week 3
Assignment 4	Week 4
Assignment 5	Week 5
Assignment 6	Week 6
Assignment 7	Week 7
Assignment 8	Week 8
Assignment 9	Week 9
Assignment 10	Week 10
Assignment 11	Week 11
Assignment 12	Week 12
Assignment 13	Week 13
Assignment 14	Week 14
Assignment 15	Week 15
Assignment 16	Week 16
Assignment 17	Week 17
Assignment 18	Week 18
Assignment 19	Week 19
Assignment 20	Week 20
Assignment 21	Week 21
Assignment 22	Week 22
Assignment 23	Week 23
Assignment 24	Week 24
Assignment 25	Week 25
Assignment 26	Week 26
Assignment 27	Week 27
Assignment 28	Week 28
Assignment 29	Week 29
Assignment 30	Week 30
Assignment 31	Week 31
Assignment 32	Week 32
Assignment 33	Week 33
Assignment 34	Week 34
Assignment 35	Week 35
Assignment 36	Week 36
Assignment 37	Week 37
Assignment 38	Week 38
Assignment 39	Week 39
Assignment 40	Week 40
Assignment 41	Week 41
Assignment 42	Week 42
Assignment 43	Week 43
Assignment 44	Week 44
Assignment 45	Week 45
Assignment 46	Week 46
Assignment 47	Week 47
Assignment 48	Week 48
Assignment 49	Week 49
Assignment 50	Week 50
Assignment 51	Week 51
Assignment 52	Week 52
Assignment 53	Week 53
Assignment 54	Week 54
Assignment 55	Week 55
Assignment 56	Week 56
Assignment 57	Week 57
Assignment 58	Week 58
Assignment 59	Week 59
Assignment 60	Week 60
Assignment 61	Week 61
Assignment 62	Week 62
Assignment 63	Week 63
Assignment 64	Week 64
Assignment 65	Week 65
Assignment 66	Week 66
Assignment 67	Week 67
Assignment 68	Week 68
Assignment 69	Week 69
Assignment 70	Week 70
Assignment 71	Week 71
Assignment 72	Week 72
Assignment 73	Week 73
Assignment 74	Week 74
Assignment 75	Week 75
Assignment 76	Week 76
Assignment 77	Week 77
Assignment 78	Week 78
Assignment 79	Week 79
Assignment 80	Week 80
Assignment 81	Week 81
Assignment 82	Week 82
Assignment 83	Week 83
Assignment 84	Week 84
Assignment 85	Week 85
Assignment 86	Week 86
Assignment 87	Week 87
Assignment 88	Week 88
Assignment 89	Week 89
Assignment 90	Week 90
Assignment 91	Week 91
Assignment 92	Week 92
Assignment 93	Week 93
Assignment 94	Week 94
Assignment 95	Week 95
Assignment 96	Week 96
Assignment 97	Week 97
Assignment 98	Week 98
Assignment 99	Week 99
Assignment 100	Week 100

To see which Arya VM is yours use the link on the class website

Accessing CIS VLab

Rich's Cabrillo College CIS Classes Home Page

Home Resources

Login
Flashcards
Admin

CIS 90
CIS 192
Previous Classes

10 days till term starts!

Cabrillo College
Web Advisor
Commands and Files

Vlab RDP file

CIS 90 VLab VM Assignments
CIS 192 VLab Pod Assignments

RJP Dennis Ritchie

Rich Simms

Contact
• Email: risimms@cabrillo.edu
• Office hours: directory page

Spring 2013 Cabrillo Linux Classes

- Introduction to UNIX/Linux (CIS 90) - Rich Simms teaching
- UNIX/Linux Network Administration (CIS 192AB) - Rich Simms teaching

Welcome to Opus
opus.cis.cabrillo.edu

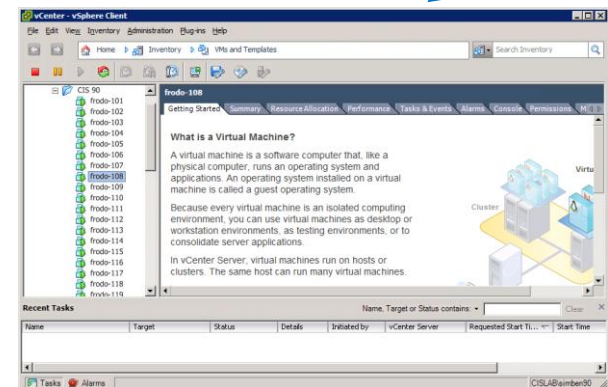
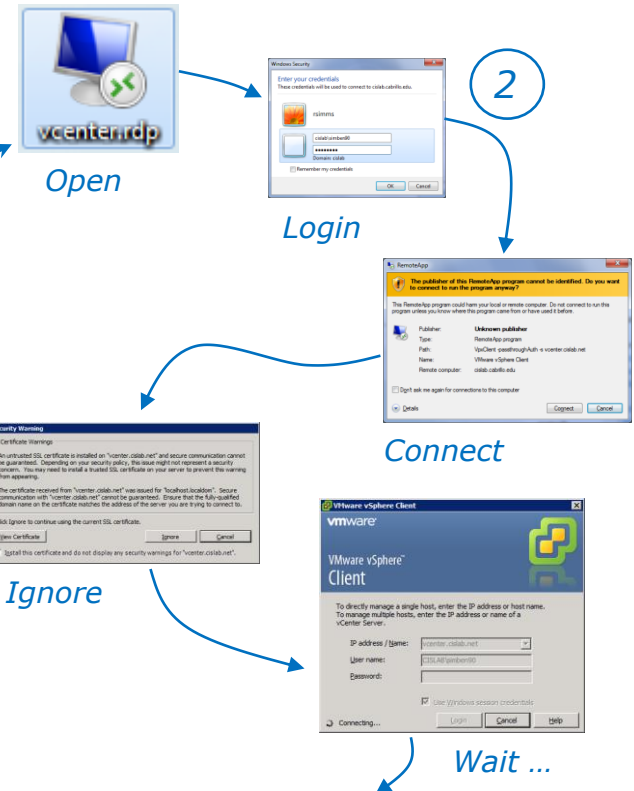
Remote access to the CIS Virtual Lab (VLab)

Download this RDP file: [vcenter.rdp](#)
(Use right-click Save As...)

1) Download the vcenter.rdp file to your desktop and then open it to access VLab.

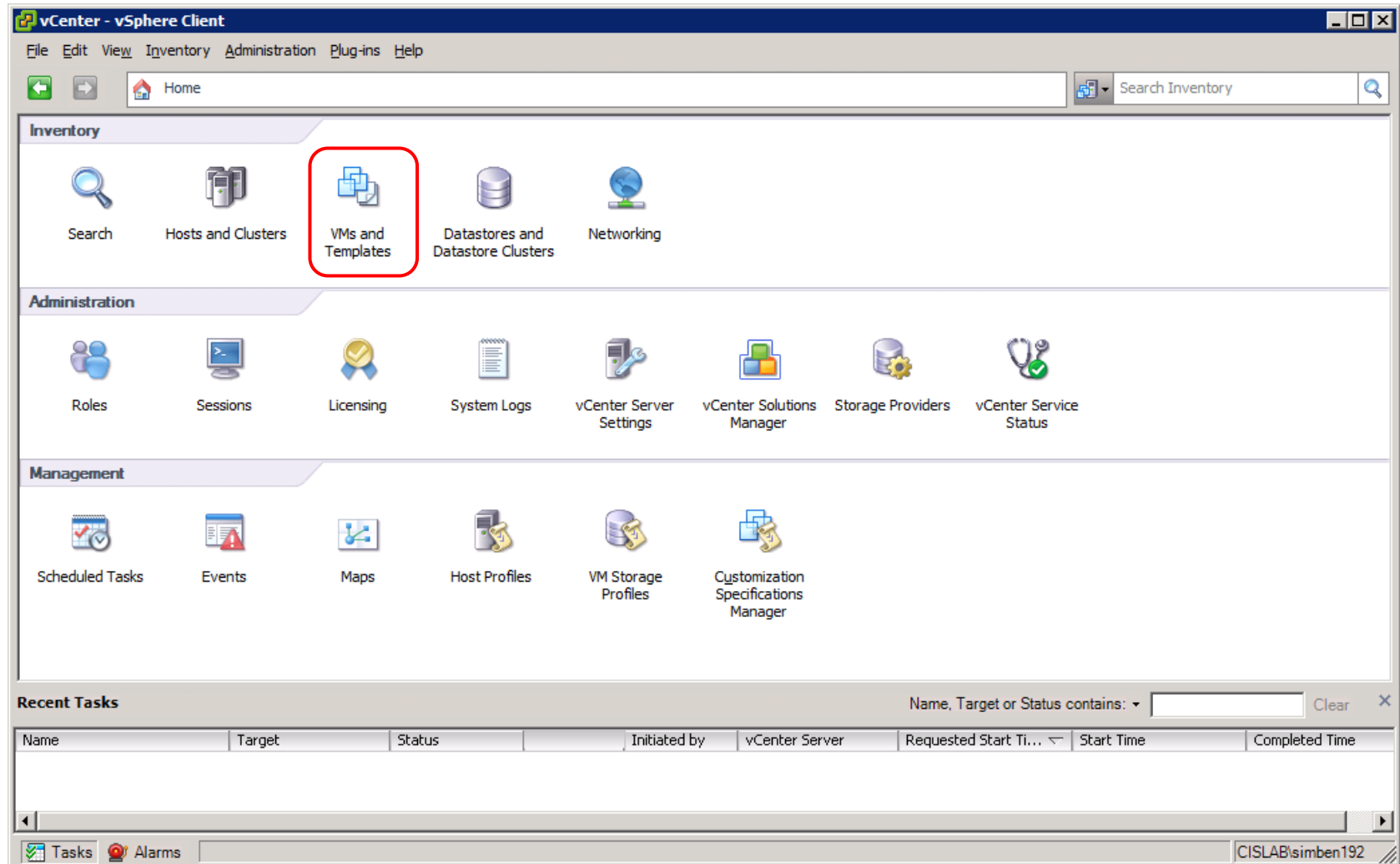
2) Mac users will **need to install** CoRD.

3) When entering your username and password you must preface your username with the "cislab", for example Benji would use: cislab\simben90



Locate and select your assigned VM

CIS VLab Home View



Click VMs and Templates to get to your course VMs

Selecting and powering on your VM

The screenshot shows the vCenter - vSphere Client interface. The left pane displays a tree view of the inventory, including 'vCenter', 'CIS VLab', 'CIS 90', and 'Student VMs'. Under 'Student VMs', a list of VMs is shown, including Arya-01 through Arya-21. Arya-04 is selected. The right pane shows the 'Getting Started' page for Arya-04, which includes a 'What is a VM' section and a 'Basic Tasks' section with a 'Shut down the virtual machine' button. A toolbar at the top contains various icons, including a 'Power On' icon (a green play button). A blue callout box points to the 'Power On' icon with the text: '2) If it is not powered on them then click the Power On icon on the toolbar. This icon will be grayed out if your VM is already running.' Another blue callout box points to Arya-04 in the VM list with the text: '1) Find and select your Arya VM'. At the bottom, the 'Recent Tasks' pane shows a table of tasks.

1) Find and select your Arya VM

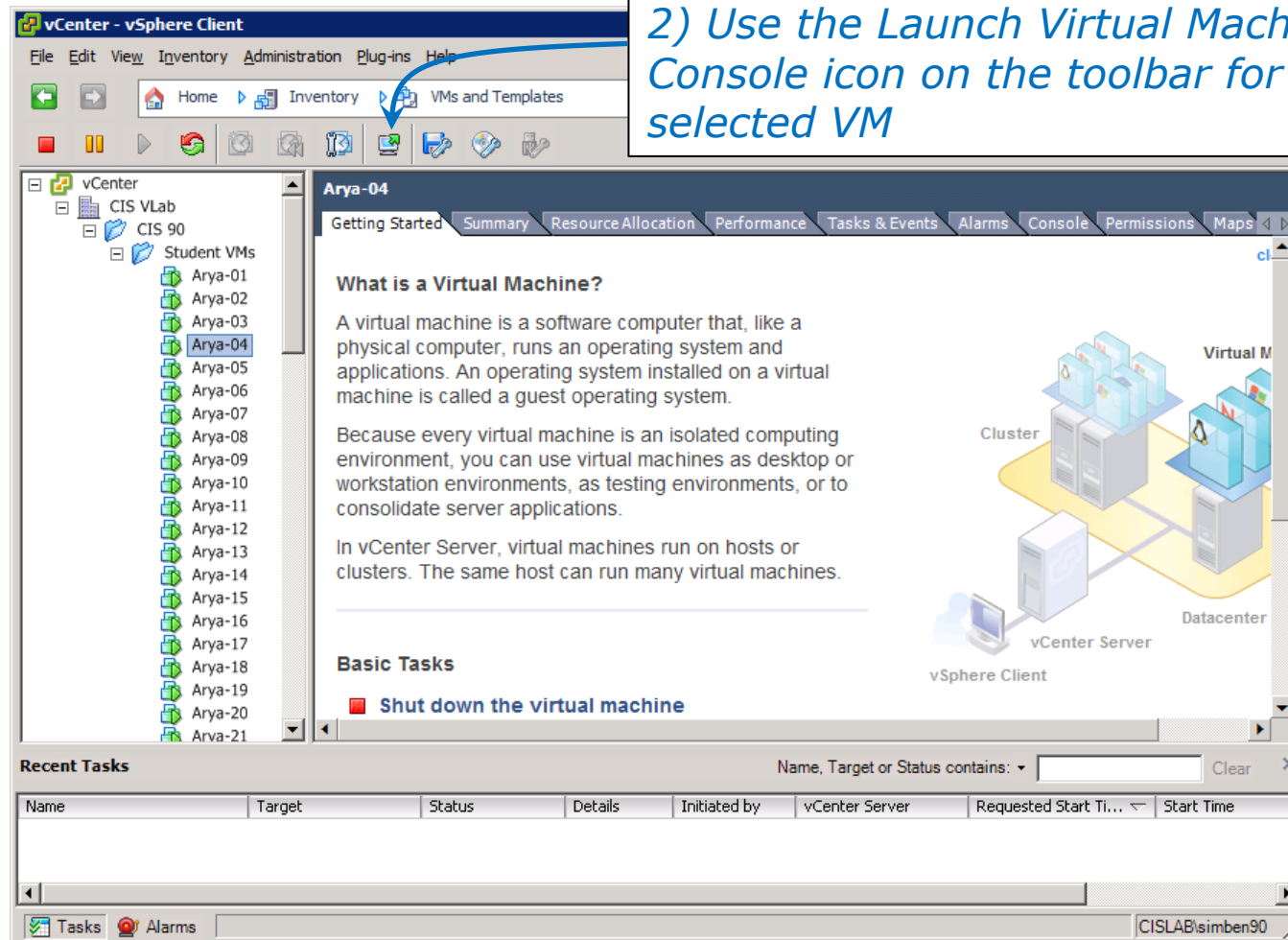
2) If it is not powered on them then click the Power On icon on the toolbar. This icon will be grayed out if your VM is already running.

Recent Tasks

Name	Target	Status	Details	Initiated by	vCenter Server	Requested Start Time	Start Time
Initiate guest OS shutd...	Arya-11	Completed		CISLAB\simb...	vCenter	8/24/2014 12:35:17 ...	8/24/2014 12:35:1...
Initiate guest OS shutd...	Arya-10	Completed		CISLAB\simb...	vCenter	8/24/2014 12:35:13 ...	8/24/2014 12:35:1...

Note that the Arya-10 and Arya-11 VMs above are not powered on

Launching a graphical console



2) Use the Launch Virtual Machine Console icon on the toolbar for the selected VM

What is a Virtual Machine?

A virtual machine is a software computer that, like a physical computer, runs an operating system and applications. An operating system installed on a virtual machine is called a guest operating system.

Because every virtual machine is an isolated computing environment, you can use virtual machines as desktop or workstation environments, as testing environments, or to consolidate server applications.

In vCenter Server, virtual machines run on hosts or clusters. The same host can run many virtual machines.

Basic Tasks

- Shut down the virtual machine

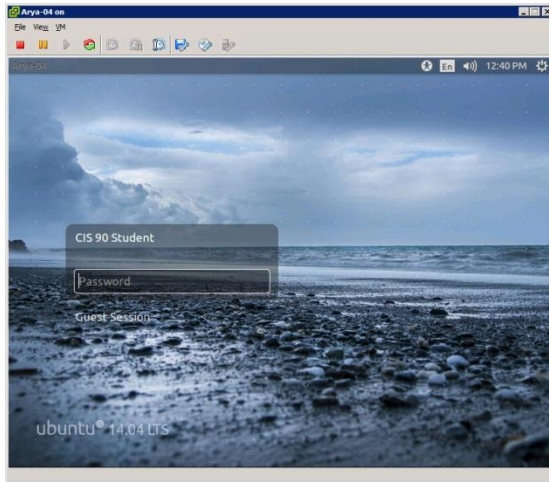
Recent Tasks

Name	Target	Status	Details	Initiated by	vCenter Server	Requested Start Ti...	Start Time

Tasks Alarms

CISLAB\simben90

Log in as
CIS 90 Student

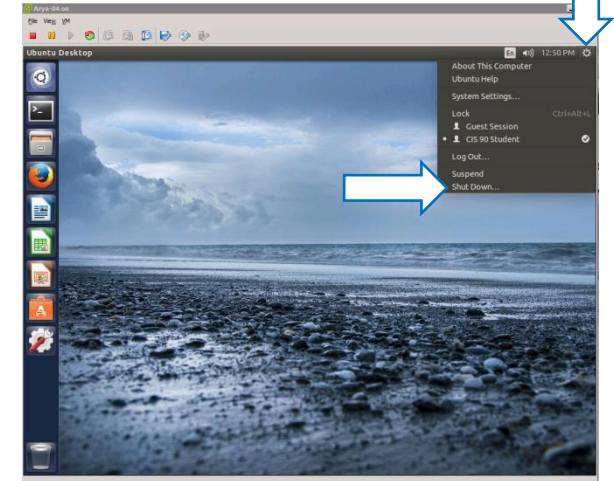


The Arya VM

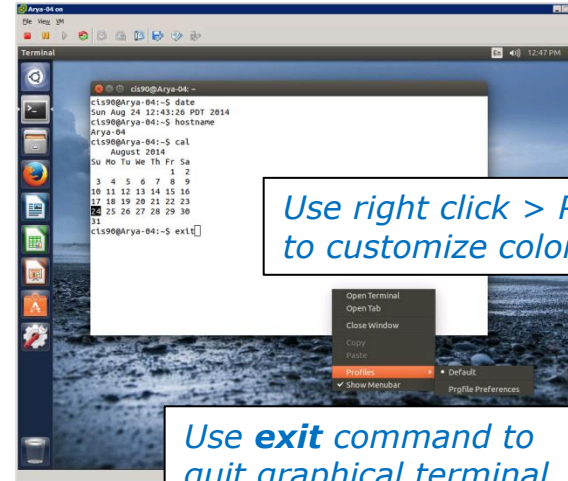
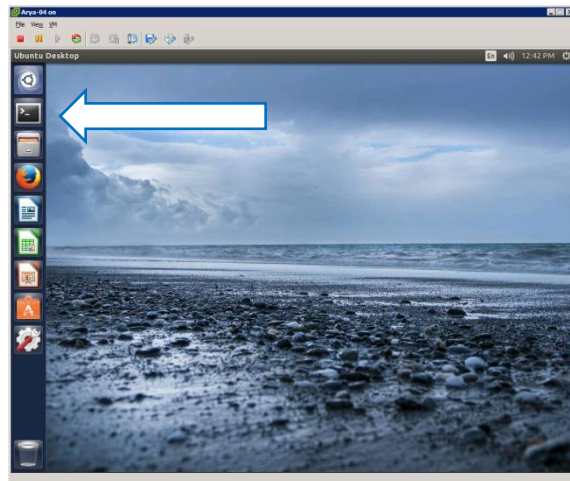


ubuntu

Shutdown using
 **> Shut Down...**



To get a graphical terminal
Terminal icon (under System Settings)



*Use right click > Profiles
to customize colors*

*Use **exit** command to
quit graphical terminal*

Command Line vs Graphical Desktop

Access the UNIX/Linux systems using:

ssh when:

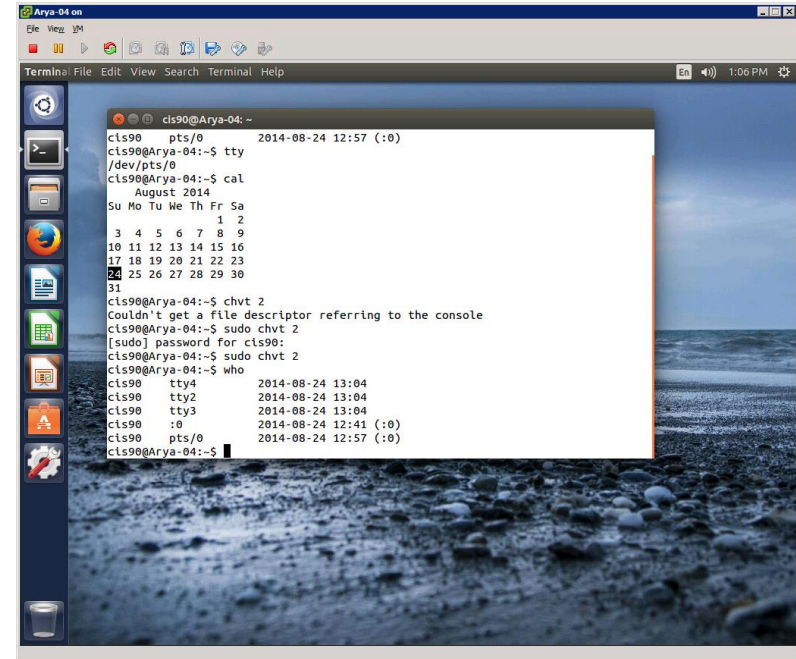
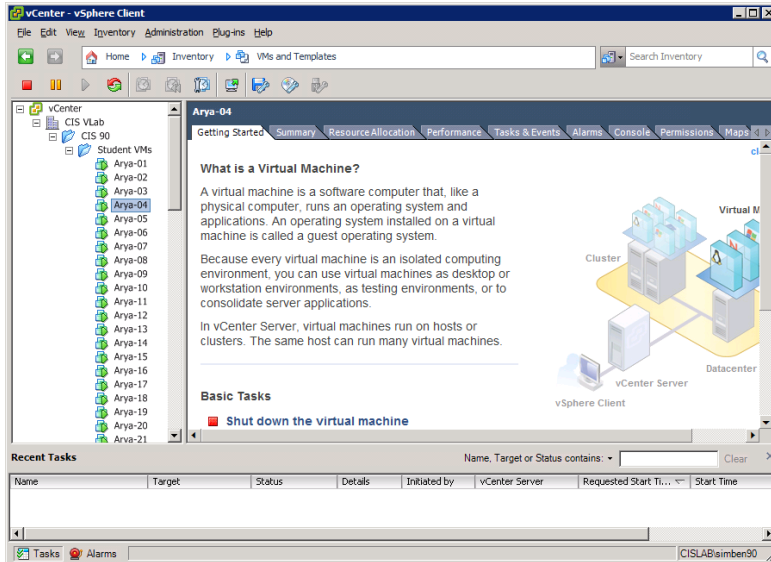
- You just need a command line
- Have a low or high speed network connection
- Note: Windows users can use Putty

VLab when:

- You want a graphical desktop
- You want to use virtual terminals (the very basic black consoles)
- Note: High speed network connection is needed
- Note: Mac users can use CoRD
- Note: you may need a fix applied to your VM if you experience the dreaded "unintended repeating key" issue

VLab = using the VMware vSphere Client via a Remote Desktop (RDP) connection

Class Activity



Try logging into CIS VLab with your **own credentials**

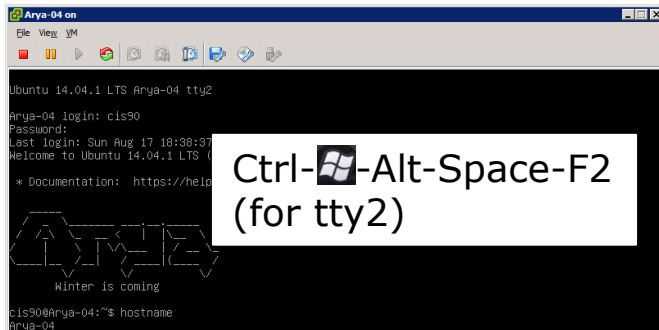
- Find your VM
- Power it on (if it's not already)
- Open a separate console for your VM
- Login as CIS 90 Student into the graphical desktop
- Run a terminal on the graphical desktop

Virtual Terminals (consoles)

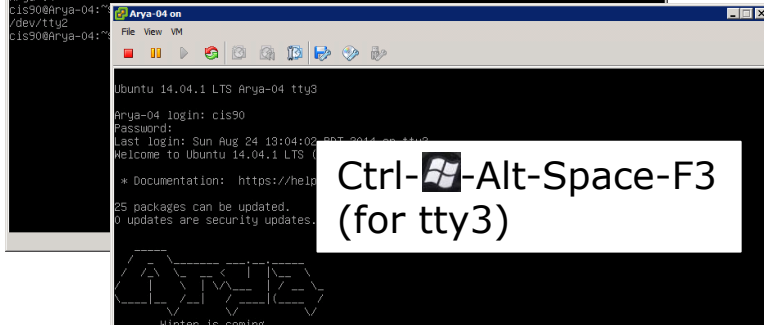
Fourth driving lesson

Virtual Terminals

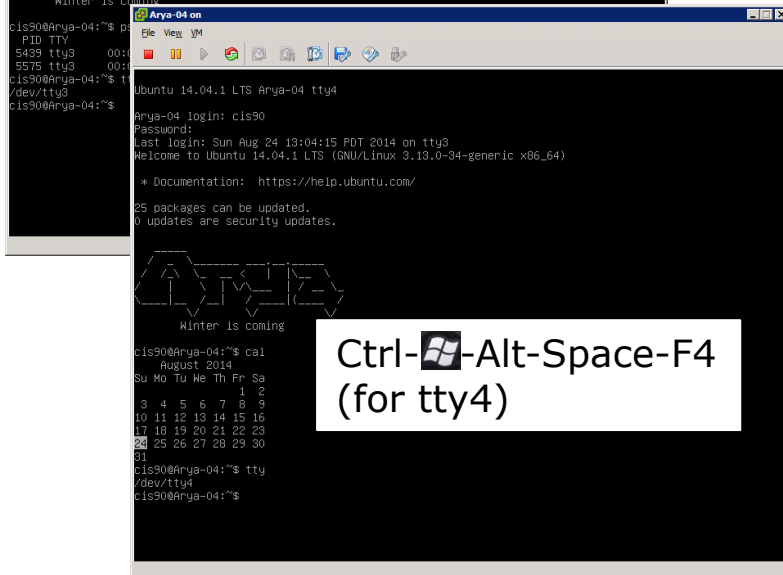
- 1) While holding down Ctrl--Alt keys, tap Space, then tap Fn key
- 2) or try: **chvt** *n*
- 3) or try: **sudo chvt** *n*
- 4) or try: **<alt-key>** *n*
(in an Ubuntu virtual terminal)



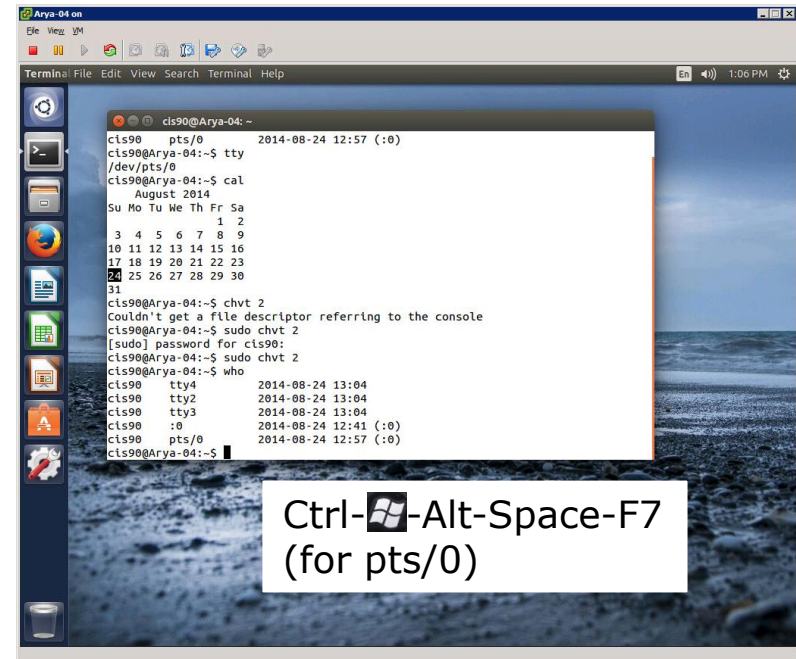
Ctrl--Alt-Space-F2
(for tty2)



Ctrl--Alt-Space-F3
(for tty3)

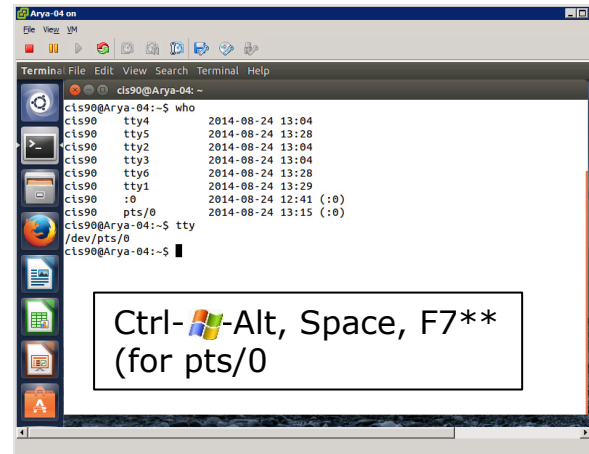
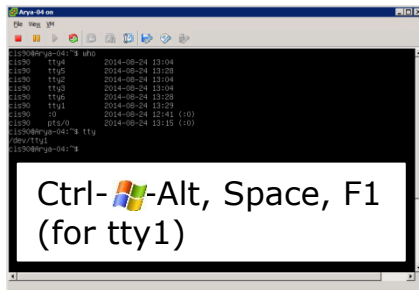


Ctrl--Alt-Space-F4
(for tty4)

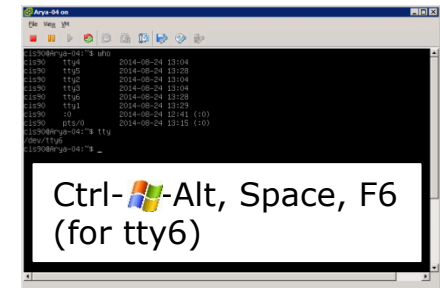


Ctrl--Alt-Space-F7
(for pts/0)

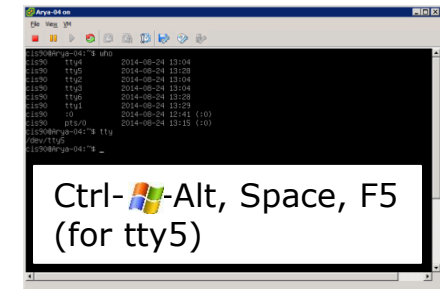
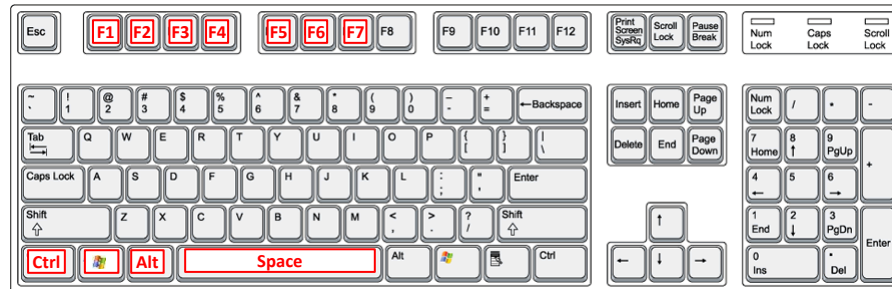
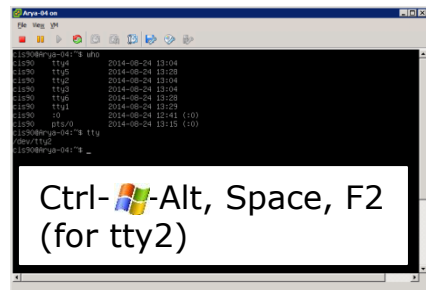
Changing Virtual TTY Terminals using **VMware vSphere**



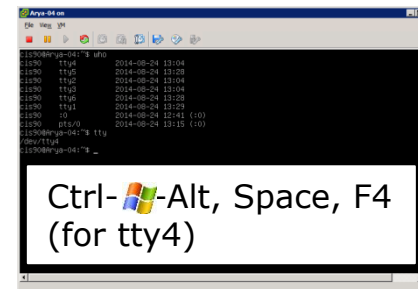
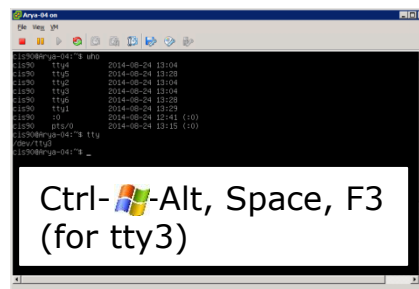
Windows PC Keyboard



While holding down Ctrl-Alt keys, tap Space, then tap Fn key*



*On some PC keyboards it is not necessary to use the key



Note: This is for vSphere only. The key and Space bar are not pressed for physical (non-VM) servers



Changing Virtual Terminals on VMware Linux VMs

VMware operations	
On PC Keyboard:	While holding down the Ctrl-  -Alt keys, tap spacebar then tap f1, f2, ... or f7.
On Mac keyboard:	Hold down Control and Option keys, tap the spacebar, hold down fn key (in addition to Control and Option keys) and tap f1, f2, ... or f7.

Pressing the  on some Windows keyboards may not be necessary

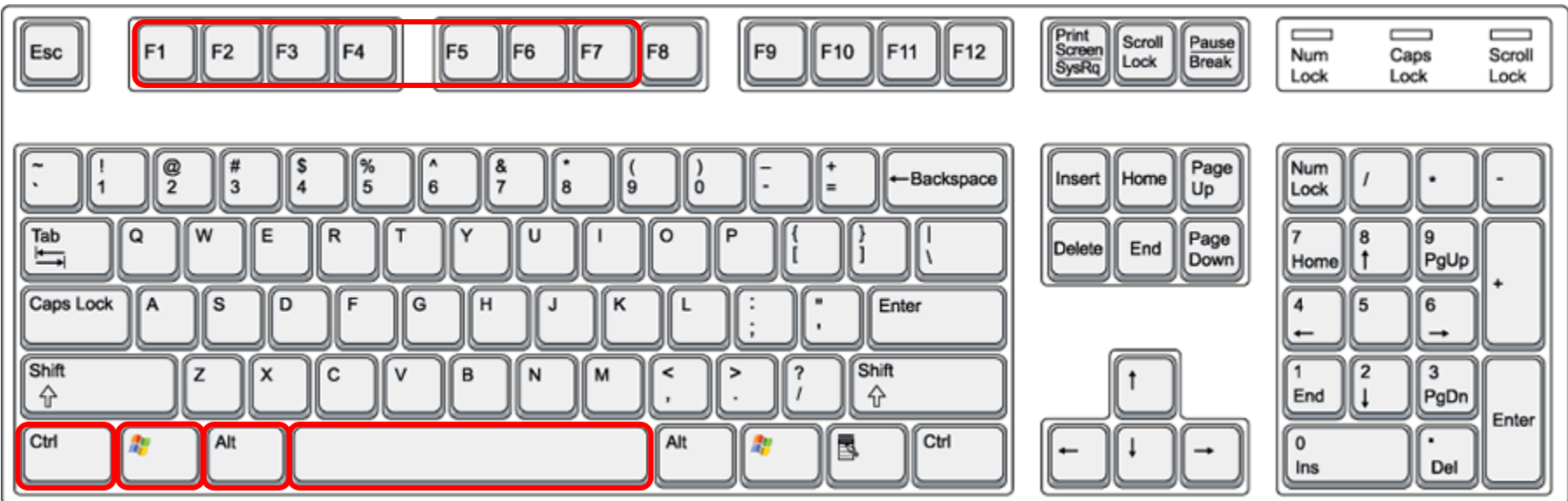
F7 is graphics mode for the Ubuntu VMs.

The Centos VMs do not have a graphics mode components installed (run level 3 only)

Note: the spacebar does not need to be tapped on a physical (non-VM) system. This is only required when changing virtual terminals on VMware VMs.

VMware VM Operations

Changing Virtual Terminals with a PC keyboard



On PC keyboard:

While holding down the **Ctrl**--**Alt** keys,
tap **Spacebar** then tap **F_n** key
(where *n*=1-7 to specify a function key)

VMware VM Operations

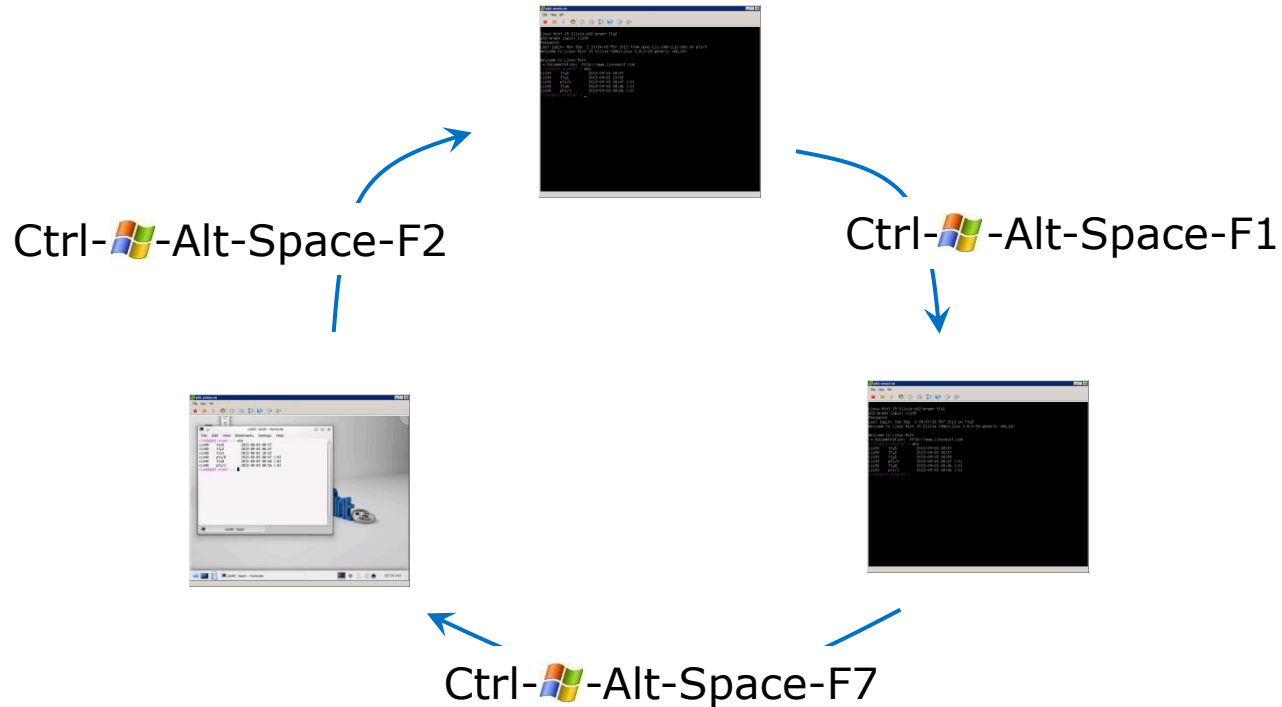
Changing Virtual Terminals with a Mac keyboard



On Mac keyboard:

While holding down the **control-option** keys
tap **Spacebar** then tap **fn-F_n** keys
(where *n*=1-7 to specify a function key)

Class Activity



On your VM:

- Try changing between the graphical desktop and the TTYs
- Login as cis90 on tty1 and tty2
- Run a terminal on the graphical desktop
- Use the who command to see how many logins there are

Terminals

Hardware Terminals



Teletype (TTY)



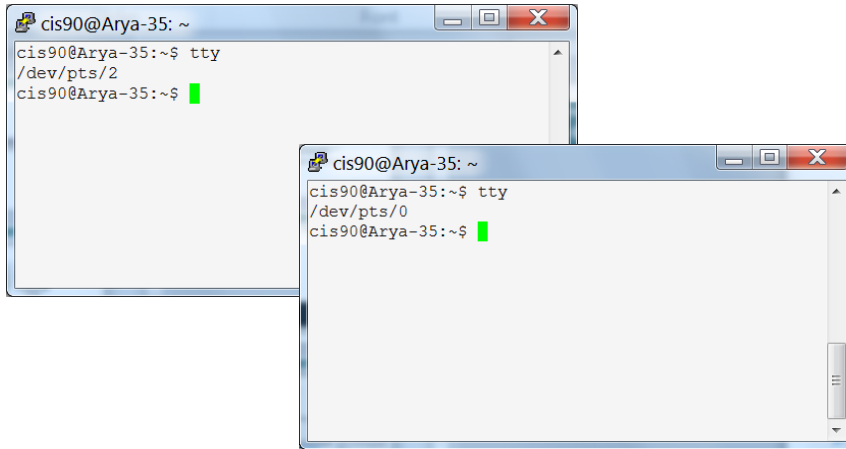
VT100



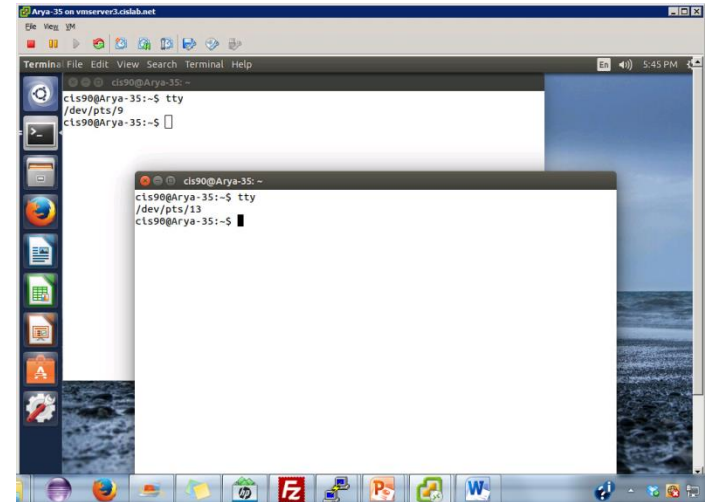
Terminals were used in the old days to interact with "minicomputers" and "mainframe" computers.

Today we use **terminal emulators** instead that are software programs.

Software Terminals



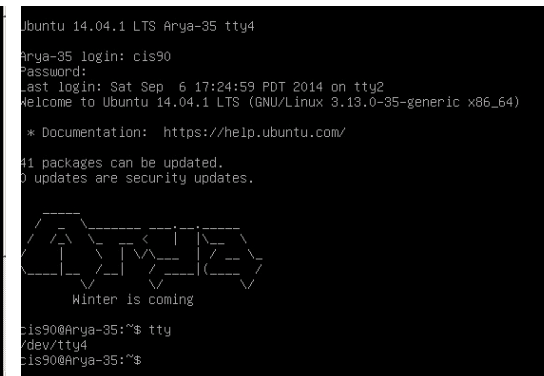
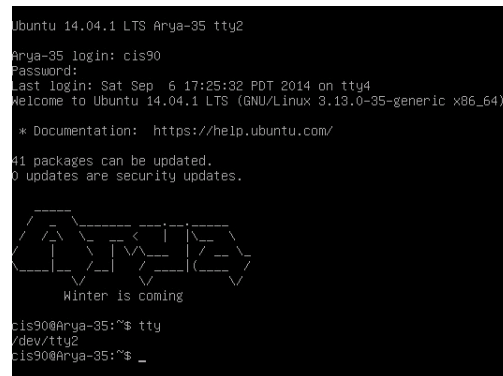
Terminal emulators like PuTTY (with scroll bars, colors, customizable backgrounds, fonts and sizes) for Windows



Graphical terminals (with scroll bars, colors, customizable backgrounds, fonts and sizes) built into Linux/Mac computers

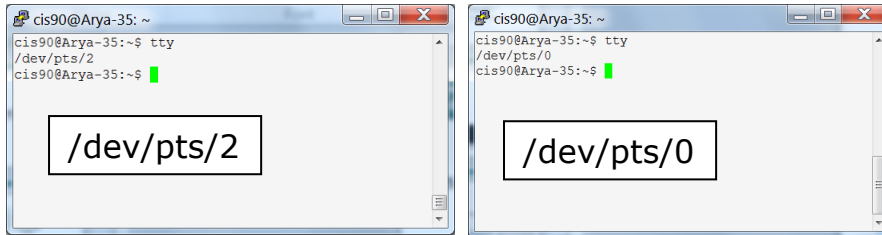
Virtual terminals (use ctrl-alt-fn)

Bare bones, no scroll bars,
also called a console



Various terminal devices on an Arya VM

Terminal emulators (e.g. Putty)



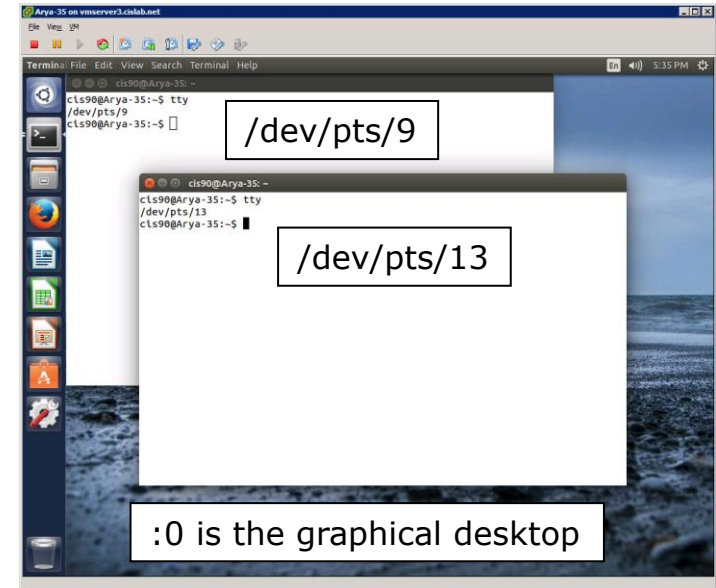
```
cis90@Arya-35:~$ who
cis90    tty4      2014-09-06 17:25
cis90    tty2      2014-09-06 17:25
cis90    pts/2     2014-09-06 17:20 (enterprise.cis.cabrillo.edu)
cis90    :0        2014-09-06 17:20 (:0)
cis90    pts/0     2014-09-06 17:21 (2601:9:6680:53b:4d09:e2b6:e7fc:d999)
cis90    pts/9     2014-09-06 17:22 (:0)
cis90    pts/13    2014-09-06 17:23 (:0)
```

pts=pseudo terminal,

tty=teletype

:n=an X window display number

Graphical terminals on graphical desktop



Virtual terminals



Putty Tips

(Note: tty = teletype)

The Putty program

```

rsimms@server0-01:~
[rsimms@server0-01 rsimms]$ ls /bin
arch      cut      fgrep    ls      pwd      sync
ash       date     gawk     mail    r        r
ash.static dd      grep     mkdir   r        r
awk       df       gtar     mknod   r        r
basename dmesg   gunzip   mktemp  r        r
bash      dnsdomainname gzip     more    r        r
bash2     doexec  hostname mount    r        r
bsh       domainname igawk    mt       s        s
cat       dumpkeys ipcalc   mv       s        s
chgrp     echo     kbd_mode netstat s        s
chmod     ed       kill     nice    s        s
chown     egrep    link     nisdomainname s
cp        env      ln       pgawk   s        s
cpio      ex       loadkeys ping     s        s
csh       false    login    ps      s        s
[rsimms@server0-01 rsimms]$

rsimms@nosmo:~/depot/gcal-3.01/src
[rsimms@nosmo src]$ ls /bin
alsaunmute  dnsdomainname  kbd_mode  nisdomainname  sync
arch         doexec         keyctl    pgawk          tar
ash          domainname    kill      ping           tcsh
ash.static   dumpkeys      ksh       ping6          touch
awk          echo          link      ps             tracepath
basename     ed            ln        pwd            tracepath6
bash         egrep         loadkeys  red            traceroute
bsh          env           login     rm             traceroute6
cat          ex            ls        rmdir          true
chgrp        false        mail      rpm            umount
chmod        fgrep        mailx     rvi            uname
chown        gawk         mkdir     rview          unicode_start
cp           gettext      mknod     sed            unicode_stop
cpio         grep         mktemp    setfont        unlink
csh          gtar         more      setserial      usleep
cut          gunzip       mount     sh             vi
date         gzip         mt        sleep          view
dd           hostname     mv        sort            ypdomainname
df           igawk        netstat   stty           zcat
dmesg        ipcalc       nice      su
[rsimms@nosmo src]$
  
```

*Why does Putty sometimes have a **black background** and sometimes a **white background**?*



Rich's Cabrillo College CIS Classes Resources

[Home](#)
[Resources](#)
[Forums](#)
[CIS Lab](#)
[CTC](#)

[Login](#)
[Flashcards](#)
[Admin](#)

[CIS 90](#)
[Previous Classes](#)

102 days till
term ends!

[Cabrillo College](#)
[Web Advisor](#)
[CCC Confer](#)
[Static IPs](#)
[Quick Ref](#)
[VM Repairs](#)
[GAH!](#)

Links

Instructors

- [Linux Master Jim](#)
- [Programming Master Ed](#)
- [Network Master Gerlinde](#)
- [Network Master Rick](#)
- [Web Master John](#)
- [Windows Master Gary](#)

Getting Linux

- [Linux ISOs](#)
- [Kernels](#)
- [RPMs \(rpmfind\)](#)
- [RPMs \(pbone\)](#)

Tools and Software

- [Apache](#)
- [Bastille](#)
- [cygwin](#)
- [DOS boot disks](#)
- [Dynamips/Dynagen](#)
- [John the Ripper](#)
- [MSDN Academic Alliance](#)
- [Netfilter](#)
- [Putty SSH Tools](#)
- [Quagga routing suite](#)
- [Tripwire](#)
- [VirtualBox](#)
- [VMware Server](#)
- [Wireshark](#)

Standards

- [IETF \(RFCs\)](#)
- [IEEE](#)

Commands

- [Practical](#)
- [Summary](#)
- [Useful](#)
- [vi summary](#)

Clubs

- [GNU Linux Users Group](#)

Departments

- [CNSA](#)
- [CIS](#)
- [CS](#)

Crib Sheets

- [Ollie Wright \(CIS 90\)](#)

Documentation

- [TLDP](#)
- [LINFO](#)

Animations

- [Linux network technologies](#)

Rich's Howtos

Putty

- [Installing PuTTY on Windows](#)
- [Configuring the appearance of PuTTY](#)

VirtualBox

- [Bringing the Eko VM home](#)

There is a Howto on the Resource page to walk you through customizing Putty

Google
(0 unread) Yah...
Cabrillo Colleg...
Rich's Cabrillo ...
Configuring th...

http://simms-teach.com/howtos/106-config-putty.html



Linux Howtos

Configuring the appearance of PuTTY

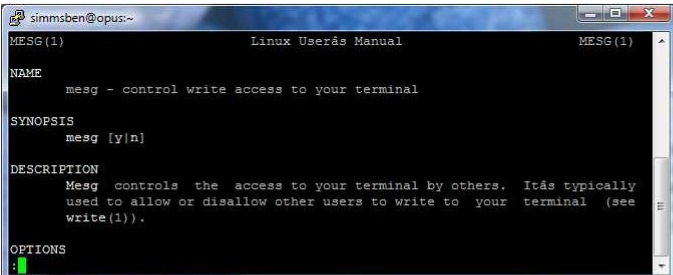
Fall, 2008

Software used

- [PuTTY SSH client \(download\)](#)

Step 1 - Run PuTTY and login

The default appearance is 10 point Courier New font with white text on a black background. The translation is ISO-8859-1 which may garble the ' displayed in "Linux User's Manual".



```
simmsben@opus:~$
MSG(1)                                Linux User's Manual                                MSG(1)
NAME
    msg - control write access to your terminal

SYNOPSIS
    msg [y|n]

DESCRIPTION
    Msg controls the access to your terminal by others. It's typically
    used to allow or disallow other users to write to your terminal (see
    write(1)).

OPTIONS
```

Right click on the top of the window to get a menu.

Step 2 - Get to Reconfiguration window





Lesson 1

Review

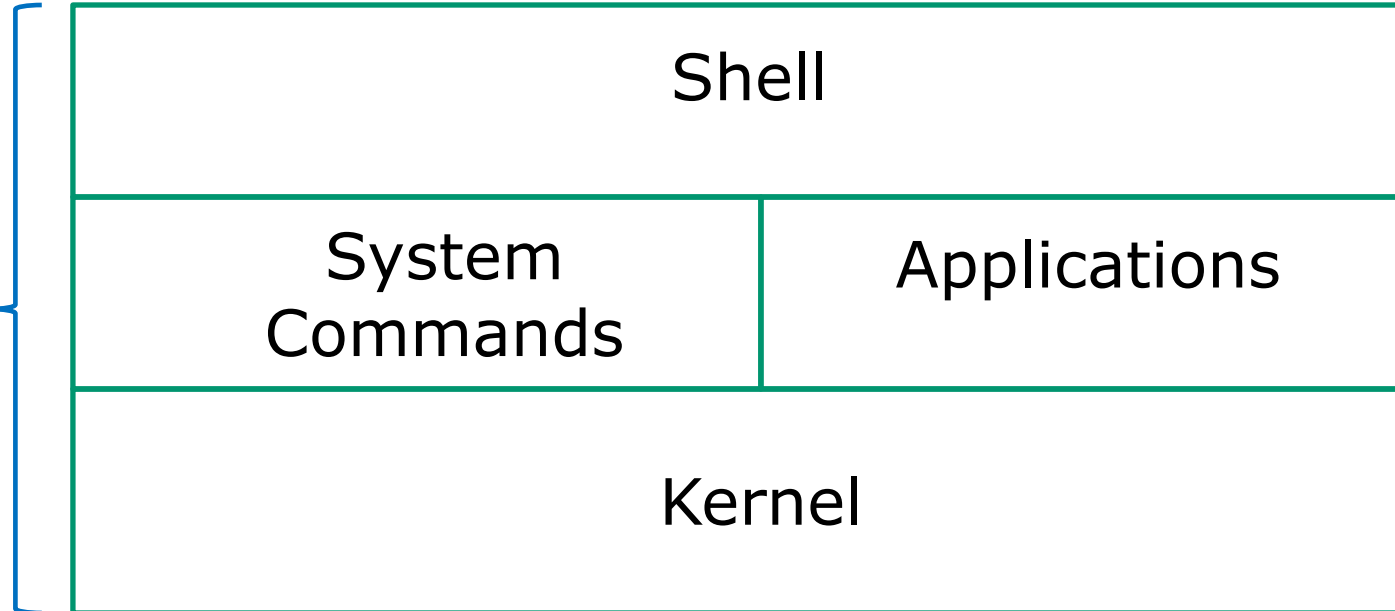
UNIX/Linux Architecture

Simplified View - Four Major Components

Users



Software



Hardware





The Lesson 1 commands for your toolbox

cal

Prints calendars

date

Shows the time and date

clear

Clears the screen

exit

Exits login session

history

Shows commands used previously

id

Shows your username and UID (and more)

ps

Shows your processes (including the name of the shell)

ssh

For connecting and logging into a remote computer

hostname

Shows the name of the computer being used

uname

Shows name of the operating system kernel

cat /etc/issue

Shows name of the "distro" (distribution)

tty

Shows which terminal device is being used

who

Shows all users who are logged in and from where

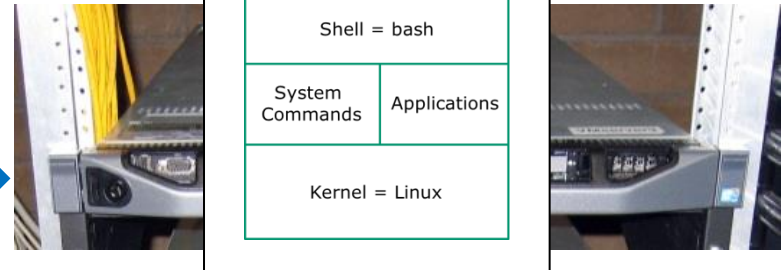
who am i

*Like **who**, but only shows your login session*

"Name" Terminology



ssh -p 2220 simben90@oslab.cishawks.net



Opus AKA **oslab.cishawks.net** AKA **oslab.cis.cabrillo.edu**

Various "names" bandied about:

User's first and last **name**: Benji Simms

user**name** = simben90

name of terminal device used = /dev/pts/2

(terminal type = xterm)

host**name** = oslab.cishawks.net

Name of distro = CentOS

Name of shell = bash

Name of kernel = Linux

To view:

/etc/passwd

id

tty

echo \$TERM

hostname

/etc/issue

ps

uname

Terminals types and devices

```
login as: simben90  
simben90@oslab.cabrillo.edu's password:  
Last login: Sat Sep  1 09:26:51 2012 from 50-0-68-  
235.dsl.dynamic.fusionbroadband.com
```

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

Hit Enter to accept

```
Welcome to Opus  
Serving Cabrillo College
```

```
Terminal type? [xterm]  
Terminal type is xterm.  
/home/cis90/simben $ tty  
/dev/pts/3
```

*The terminal type is **xterm***

*The terminal device for this session is **/dev/pts/3***

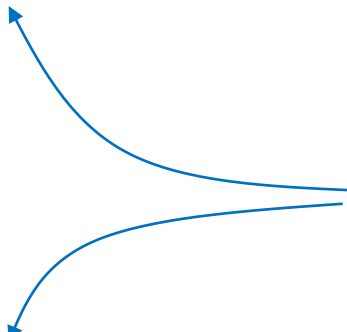
The **terminal type** is not the same as the **terminal device**

How can I print a calendar?

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

```
September 2012
Su Mo Tu We Th Fr Sa
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
```

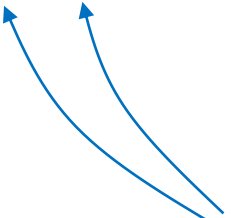
*The **cal** command*



```
/home/cis90/simben $ cal 9 2001
```

```
September 2001
Su Mo Tu We Th Fr Sa
                1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
```

*Month and year **arguments***



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

A command can have arguments

What is the current time and date?

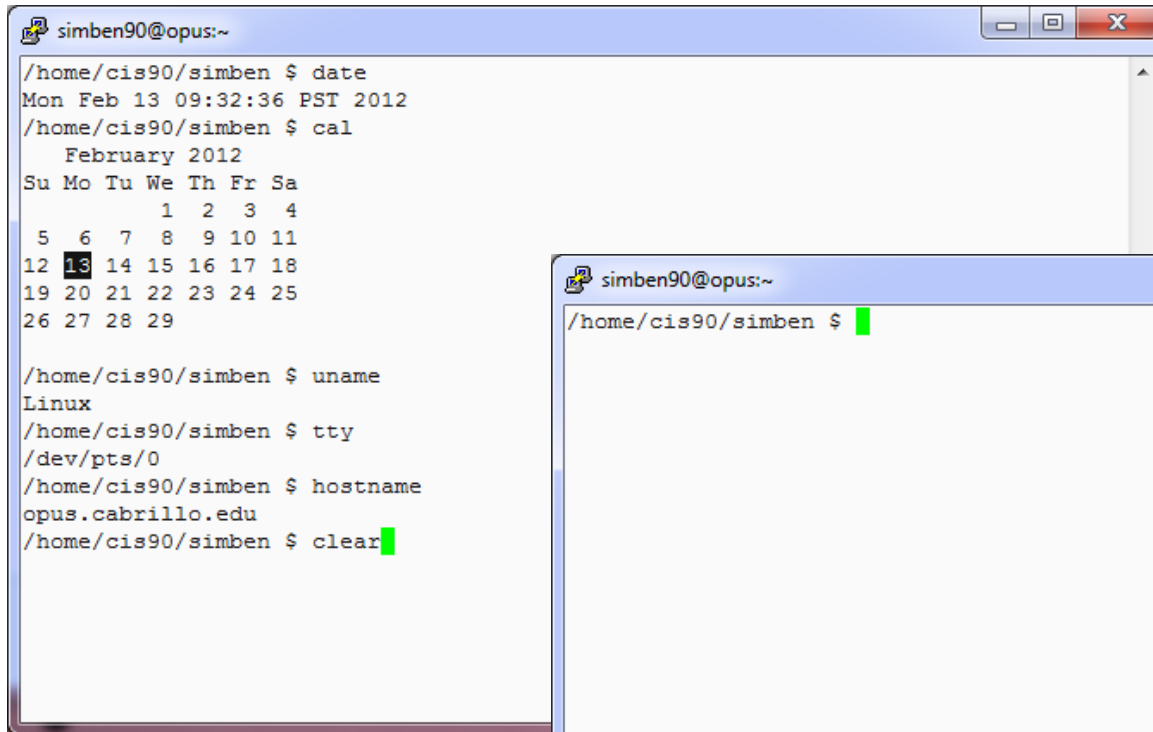
The shell "prompt"

The "command"

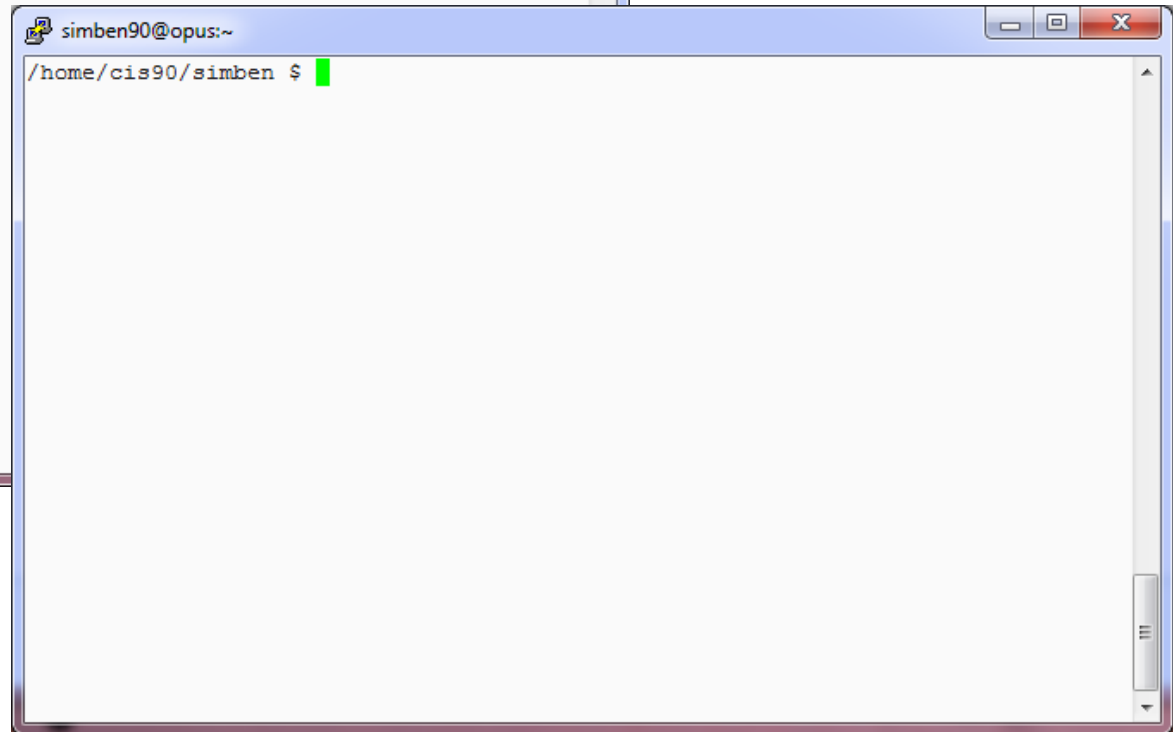
```
/home/cis90/simben $ date  
Sat Sep  1 14:03:33 PDT 2012  
/home/cis90/simben $
```

The prompt is output by the shell, you type the command

How do I clear the screen?



```
simben90@opus:~  
/home/cis90/simben $ date  
Mon Feb 13 09:32:36 PST 2012  
/home/cis90/simben $ cal  
February 2012  
Su Mo Tu We Th Fr Sa  
      1  2  3  4  
 5  6  7  8  9 10 11  
12 13 14 15 16 17 18  
19 20 21 22 23 24 25  
26 27 28 29  
  
/home/cis90/simben $ uname  
Linux  
/home/cis90/simben $ tty  
/dev/pts/0  
/home/cis90/simben $ hostname  
opus.cabrillo.edu  
/home/cis90/simben $ clear
```

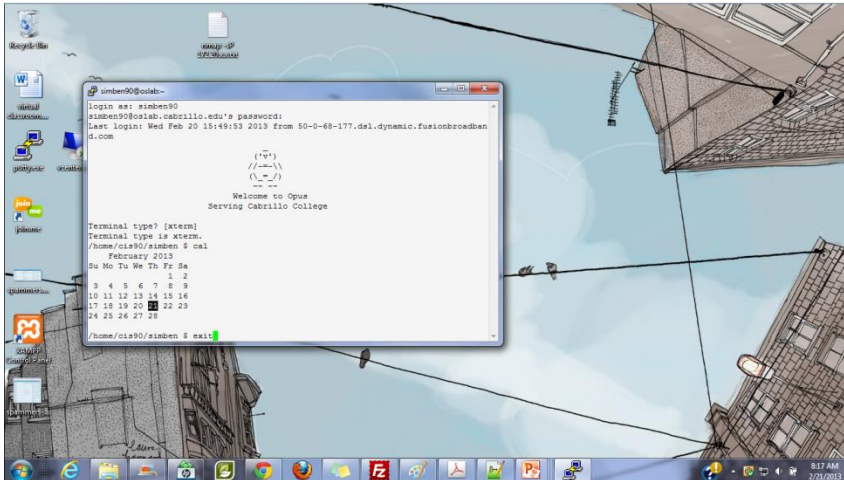


```
simben90@opus:~  
/home/cis90/simben $
```

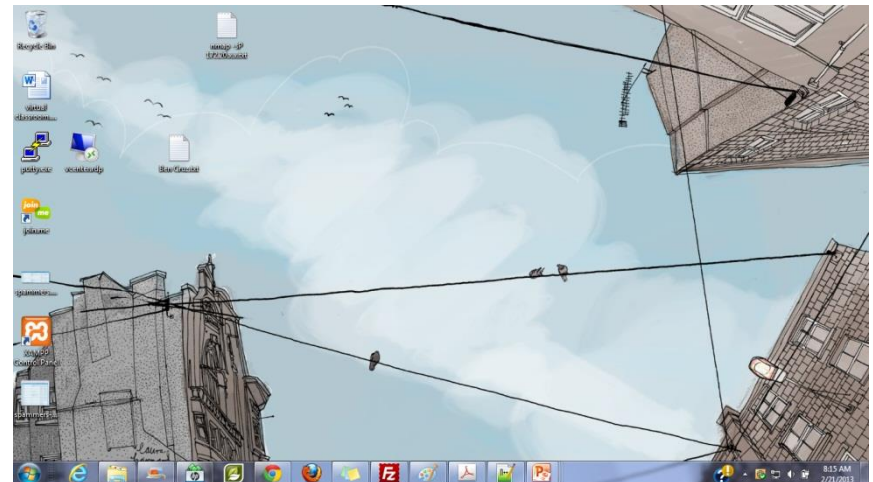
The **clear** command scrolls previous commands out of sight

How do I end this login session?

before **exit**



after **exit**



The **exit** command ends the session and the terminal window disappears ... POOF!

Viewing your command history

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

```
1  hostname  
2  exit  
3  who  
4  who -q  
5  ps -e
```

```
< snipped >
```

```
177 cal 9 2001  
178 exit  
179 who  
180 cal  
181 tty  
182 uname  
183 ps  
184 id  
185 exit  
186 history
```

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

*The **history** command outputs the commands used previously ... even from previous login sessions*

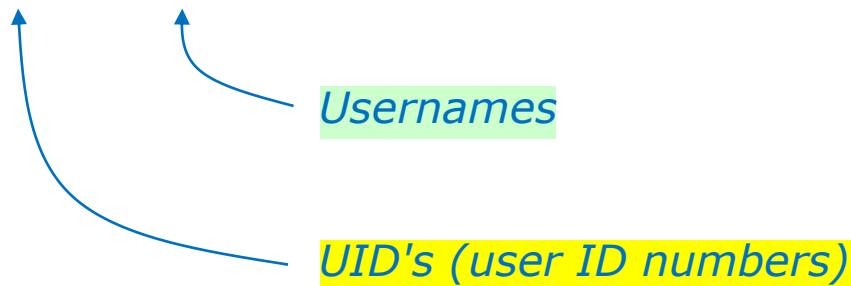
Tip: Use the "Up Arrow" key to quickly re-issue a previous command!

What is the UID (User ID) for my account or other accounts?

```
/home/cis90/simben $ id
uid=1001(simben90) gid=190(cis90) groups=190(cis90),100(users)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

```
/home/cis90/simben $ id milhom90
uid=1002(milhom90) gid=190(cis90) groups=190(cis90),100(users)
```

```
/home/cis90/simben $ id simben90
uid=1001(simben90) gid=190(cis90) groups=190(cis90),100(users)
```



We are all just numbers to the Linux kernel

What shell am I using?

Shell	
System Commands	Applications
Kernel	

```

/home/cis90/simben $ ps
  PID TTY          TIME CMD
 28994 pts/0    00:00:00 bash
 29093 pts/0    00:00:00 ps
  
```

Process ID numbers →

Terminal device being used →

*the shell is sleeping and waiting for **ps** command to finish* →

***ps** command is running as it outputs this* →

The **ps** command outputs the current processes you own including the shell program you are using

How do I log into another computer system?

Method 1: The **ssh** command using a hostname

username on remote computer → *Hostname of remote computer*

```
/home/cis90/simben $ ssh cis90@p06-arwen
cis90@p06-arwen's password:
Welcome to Linux Mint 15 Olivia (GNU/Linux 3.8.0-26-generic x86_64)

Welcome to Linux Mint
* Documentation: http://www.linuxmint.com
Last login: Sun Sep  8 09:52:00 2013
cis90@p06-arwen:~ >
```

Notice how the prompt changes on the remote computer →

*Note: You can also **ssh** into the same computer you are currently using for an additional session.*

How do I log into another computer system?

Method 1: The **ssh** command using an IP address

username on remote computer

IP address of remote computer

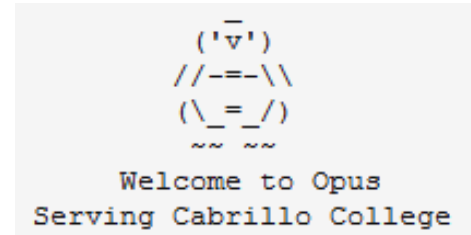
```
/home/cis90/simben $ ssh cis90@172.20.4.34  
cis90@172.20.4.34's password:  
Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-generic x86_64)
```

*Notice how
the prompt
changes on
the remote
computer*

```
* Documentation:  https://help.ubuntu.com/  
  
361 packages can be updated.  
109 updates are security updates.  
  
Last login: Wed Feb 20 17:26:25 2013 from oslab.cabrillo.edu  
cis90@frodo-108:~$
```

What is the name of the computer I'm interacting with?

```
/home/cis90/simben $ hostname  
oslab.cishawks.net
```

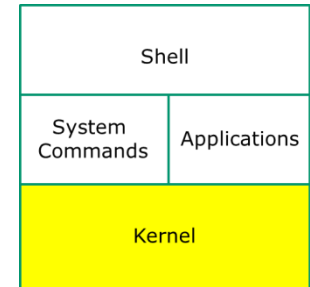


We still refer to Opus as "Opus" in this class however it's official hostname on the Internet is "oslab". This may change in the future after some network changes are made.

Opus is a member of two overlapping Internet domains:

- The **cis.cabrillo.edu** domain is a sub-domain of the college's domain.
- The **cishawks.net** domain is an alternate domain put in place to alleviate some DNS issues experienced during the CIS Lab move to building 800.

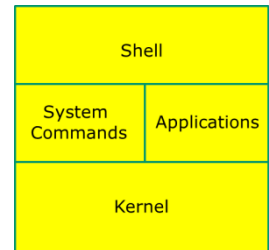
What kernel am I running on?



```
/home/cis90/simben $ uname  
Linux
```

The **uname** command (with no arguments) outputs the name of the operating system kernel

What "distro" has been installed?



```
/home/cis90/simben $ cat /etc/issue
CentOS release 6.2 (Final)
Kernel \r on \l
```

```
/home/cis90/simben $ cat /etc/*-release
CentOS release 6.2 (Final)
CentOS release 6.2 (Final)
CentOS release 6.2 (Final)
```

Catting out these files *usually* will show the distro name



What terminal device am I using?

```
/home/cis90/simben $ tty  
/dev/pts/5
```

The **terminal type** is not the same as the **terminal device**

Who else is logged in and from where?

```
/home/cis90/simben $ who
simben90 pts/0      2013-02-21 08:17 (50-0-68-28.dsl.dynamic.fusion.com)
simben90 pts/1      2013-02-21 08:45 (50-0-68-28.dsl.dynamic.fusion.com)
milhom90 pts/2      2013-02-21 08:46 (50-0-68-28.dsl.dynamic.fusion.com)
rsimms    pts/4      2013-02-21 08:46 (50-0-68-28.dsl.dynamic.fusion.com)
rodduk90  pts/7      2013-02-21 08:46 (50-0-68-28.dsl.dynamic.fusion.com)
simben90  pts/8      2013-02-21 08:49 (172.20.4.34)
milhom90  pts/9      2013-02-21 08:50 (sun-hwa.cislab.net)
```

username

*terminal device
(pts/5 = /dev/pts/5)*

*where they logged
in from (hostname
or IP address)*

The who command shows who is logged in, their terminal device, when they logged in and from where they logged in

Which is my login session?

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

```
simben90 pts/0      2013-02-21 08:17 (50-0-68-28.dsl.dynamic.fusion.com)
simben90 pts/1      2013-02-21 08:45 (50-0-68-28.dsl.dynamic.fusion.com)
milhom90 pts/2      2013-02-21 08:46 (50-0-68-28.dsl.dynamic.fusion.com)
rsimms    pts/4      2013-02-21 08:46 (50-0-68-28.dsl.dynamic.fusion.com)
rodduk90 pts/7      2013-02-21 08:46 (50-0-68-28.dsl.dynamic.fusion.com)
simben90 pts/8      2013-02-21 08:49 (172.20.4.34)
milhom90 pts/9      2013-02-21 08:50 (sun-hwa.cislab.net)
```

```
/home/cis90/simben $ who am i
```

```
simben90 pts/0      2013-02-21 08:17 (50-0-68-177.dsl.dynamic.fusion.com)
```

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

```
/dev/pts/0
```

When logged in multiple times use the terminal device to distinguish the sessions

Test your knowledge

What's the name of the terminal device I'm using right now?

```
login as: simben90
simben90@oslab.cabrillo.edu's password:
Last login: Sat Sep  1 09:26:51 2012 from 172.30.90.83
```

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

```
      Welcome to Opus
    Serving Cabrillo College
```

```
Terminal type? [xterm]
Terminal type is xterm.
/home/cis90/simben $
```

What's the name of the terminal device I'm using right now?

```
login as: simben90
simben90@oslab.cabrillo.edu's password:
Last login: Sat Sep  1 09:26:51 2012 from 172.30.90.83
```

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

```
Welcome to Opus
Serving Cabrillo College
```

```
Terminal type? [xterm]
Terminal type is xterm.
/home/cis90/simben $
/home/cis90/simben $ tty
/dev/pts/0
/home/cis90/simben $
```

Answer: /dev/pts/0

*Use the **tty** command
to find out*

What type of terminal am I using right now?

```
login as: simben90
simben90@oslab.cabrillo.edu's password:
Last login: Sat Sep  1 09:26:51 2012 from 172.30.90.83
```

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

```
      Welcome to Opus
    Serving Cabrillo College
```

```
Terminal type? [xterm]
Terminal type is xterm.
/home/cis90/simben $
```

What type of terminal am I using right now?

```
login as: simben90
simben90@oslab.cabrillo.edu's password:
Last login: Sat Sep  1 09:26:51 2012 from 172.30.90.83
```

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

```
      Welcome to Opus
    Serving Cabrillo College
```

```
Terminal type? [xterm]
Terminal type is xterm.
/home/cis90/simben $
```

Answer: xterm

We have the answer already!

What is the hostname of the computer I'm using?

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


What is the hostname of the computer I'm using?

```
/home/cis90/simben $  
/home/cis90/simben $ hostname  
oslab.cabrillo.edu  
/home/cis90/simben $
```

Answer: oslab.cabrillo.edu

*Use the **hostname**
command to find out*

What is the name of the OS (operating System) kernel?

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

What is the name of the OS (operating System) kernel?

```
/home/cis90/simben $  
/home/cis90/simben $ uname  
Linux  
/home/cis90/simben $
```

*Use the **uname**
command to find out*

Answer: Linux

What is the name of the Linux Distribution being run?

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

What is the name of the Linux Distribution being run?

```
/home/cis90/simben $ cat /etc/issue
```

```
CentOS release 6.2 (Final)
```

```
Kernel \r on \l
```

```
/home/cis90/simben $ cat /etc/*-release
```

```
CentOS release 6.2 (Final)
```

```
CentOS release 6.2 (Final)
```

```
CentOS release 6.2 (Final)
```

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

Answer: CentOS

*Use either **cat /etc/issue** or **cat /etc/*-release** to find out*

What is my username and uid (user ID number)?

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

What is my username and uid (user ID number)?

```
/home/cis90/simben $  
/home/cis90/simben $ id  
uid=1001(simben90) gid=190(cis90)  
groups=190(cis90),100(users)  
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
/home/cis90/simben $
```

Answer: username=simben90 and the uid=1001

*Use the **id** command
to find out*

What is the name of the shell I'm using?

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


What is the name of the shell I'm using?

```
/home/cis90/simben $  
/home/cis90/simben $ ps  
  PID TTY          TIME CMD  
28237 pts/0    00:00:00 bash  
28752 pts/0    00:00:00 ps  
/home/cis90/simben $
```

Answer: bash

*Use the **ps** command to find out.*

We will soon learn another command for doing this.