Lesson Module Checklist

- Slides
- Flash cards
- First minute quiz
- Web calendar summary
- Web book pages
- Commands
- Howtos

- Lab tested
- Opus - submit and turnin directory tested

- Bring Add Codes
- Bring printed roster

- Backup slides, Whiteboard slides, handouts on flash drive
- 9V backup battery for microphone
- Key card for door

Student checklist

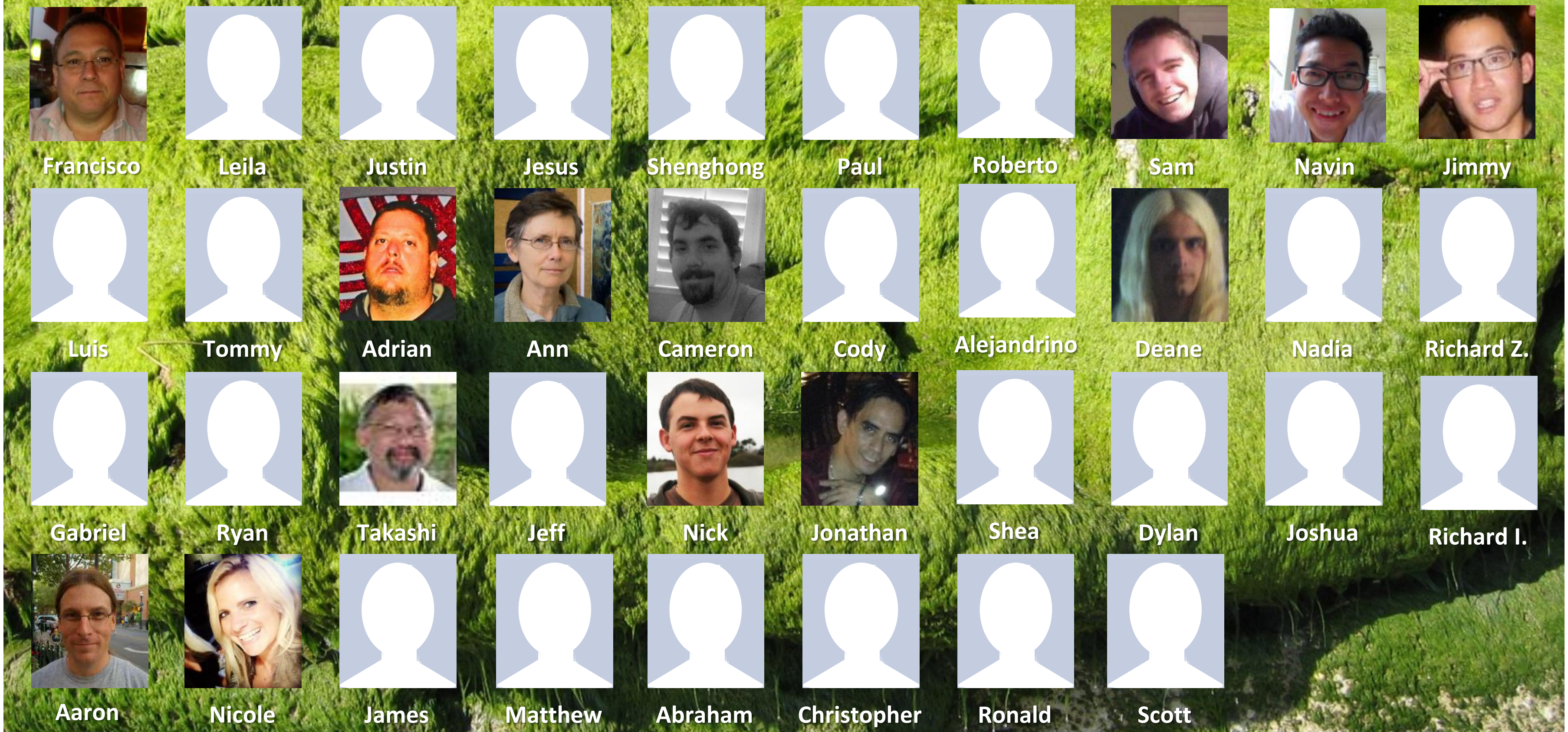
1) Browse to the CIS 90 website Calendar page
   - http://simms-teach.com
   - Click CIS 90 link on left panel
   - Click Calendar link near top of content area
   - Locate today's lesson on the Calendar

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

3) Click Enter virtual classroom to join CCC Confer session

4) Connect to Opus using Putty or ssh command

Instructor: **Rich Simms**
Dial-in: **888-886-3951**
Passcode: **136690**



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

Introductions and Credits

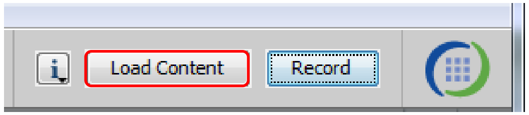
Jim Griffin
- Created this Linux course
- Created Opus and the CIS VLab
- Jim’s site: http://cabrillo.edu/~jgriffin/

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

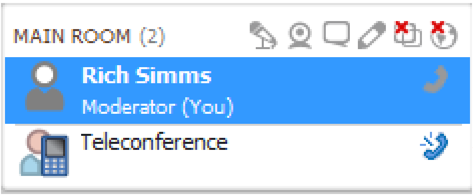
And thanks to:
- John Govsky for many teaching best practices: e.g. the First Minute quizzes, the online forum, and the point grading system (http://teacherjohn.com/)

Instructor CCC Confer checklist

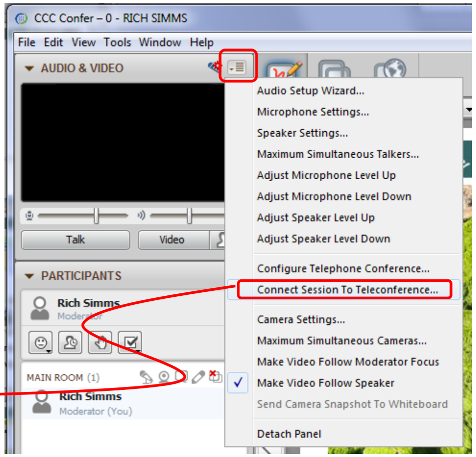
[ ] Preload White Board



[ ] Connect session to Teleconference

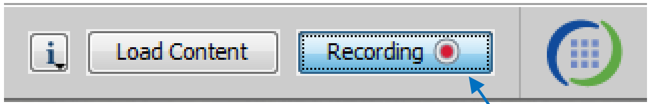


Session now connected to teleconference



CCC Confer – 0 - RICH SIMMS
File Edit View Tools Window Help
AUDIO & VIDEO
Audio Setup Wizard...
Microphone Settings...
Speaker Settings...
Maximum Simultaneous Talkers...
Adjust Microphone Level Up
Adjust Microphone Level Down
Adjust Speaker Level Up
Adjust Speaker Level Down
Configure Telephone Conference...
Connect Session To Teleconference...
Camera Settings...
Maximum Simultaneous Cameras...
Make Video Follow Moderator Focus
✓ Make Video Follow Speaker
Send Camera Snapshot To Whiteboard
Detach Panel
Talk
Video
PARTICIPANTS
Rich Simms
Moderator
MAIN ROOM (1)
Rich Simms
Moderator (You)

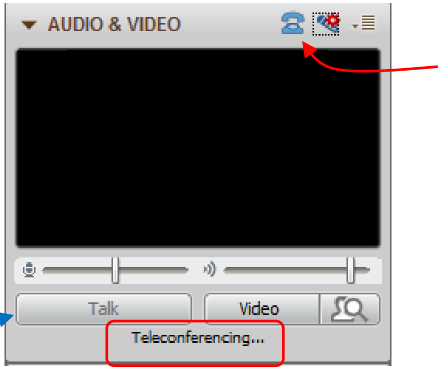
[ ] Is recording on?



Red dot means recording

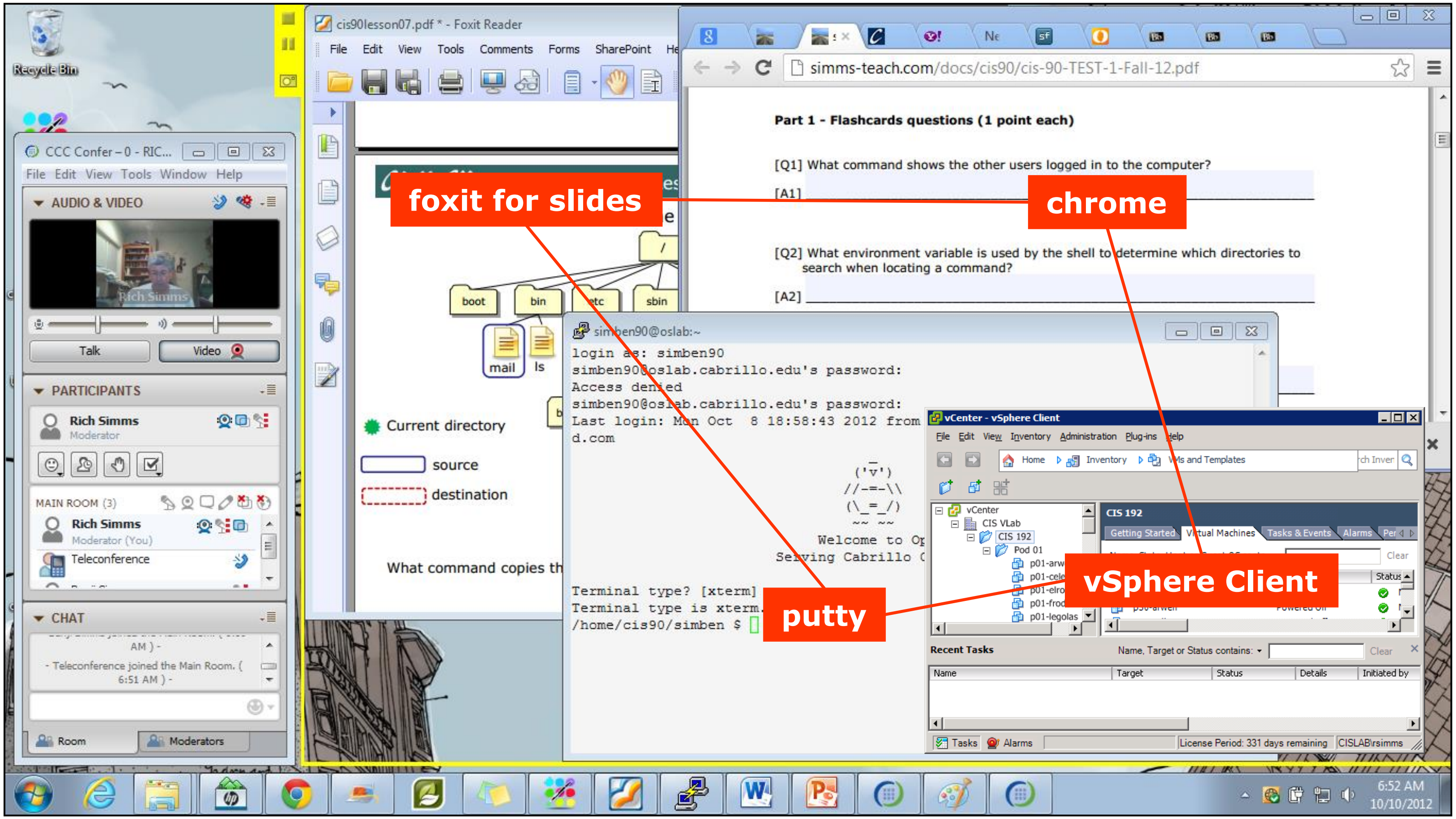
[ ] Use teleconferencing, not mic

Should be greyed out



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

Instructor CCC Confer checklist



foxit for slides

chrome

putty

vSphere Client

simms-teach.com/docs/cis90/cis-90-TEST-1-Fall-12.pdf

Part 1 - Flashcards questions (1 point each)

[Q1] What command shows the other users logged in to the computer?

[A1] ______________________________

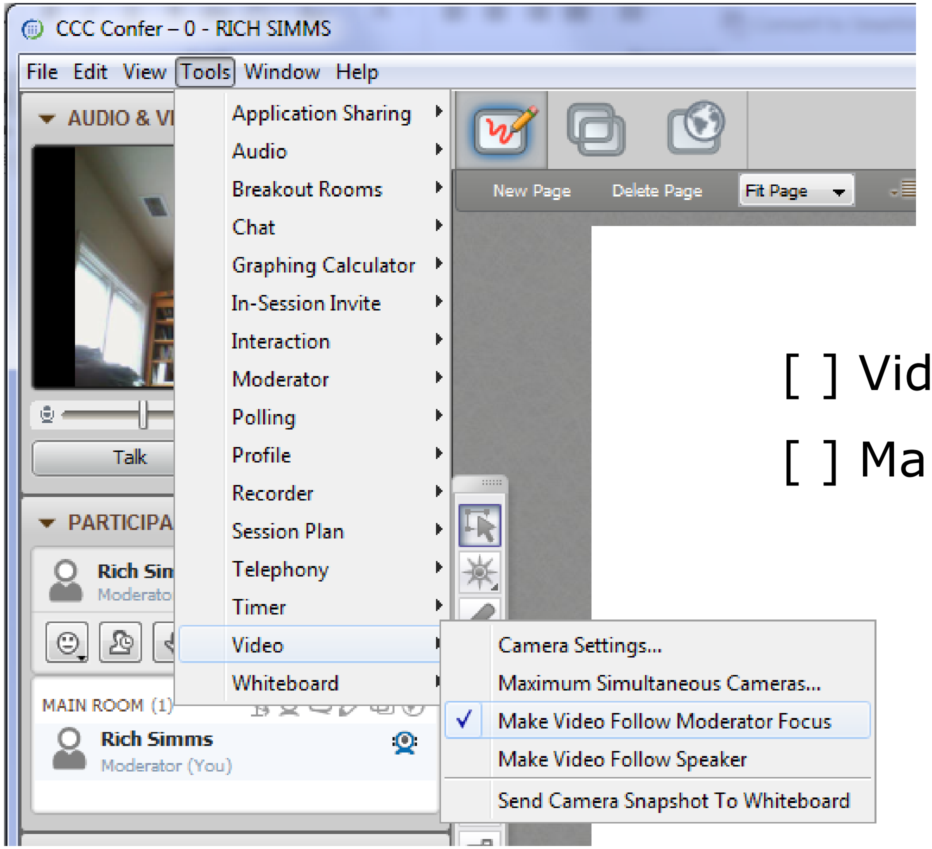
[Q2] What environment variable is used by the shell to determine which directories to search when locating a command?

[A2] ______________________________

```
login as: simben90
simben90@oslab.cabrillo.edu's password:
Access denied
simben90@oslab.cabrillo.edu's password:
Last login: Mon Oct  8 18:58:43 2012 from
d.com

                                _
                              ('v')
                             //-=-\\
                             (\_=_/)
                              ~~ ~~
                         Welcome to Op
                    Serving Cabrillo C

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

[ ] layout and share apps

Instructor CCC Confer checklist



✓ Make Video Follow Moderator Focus

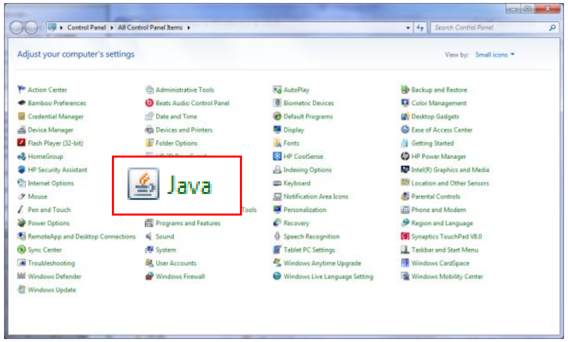
[ ] Video (webcam)

[ ] Make Video Follow Moderator Focus

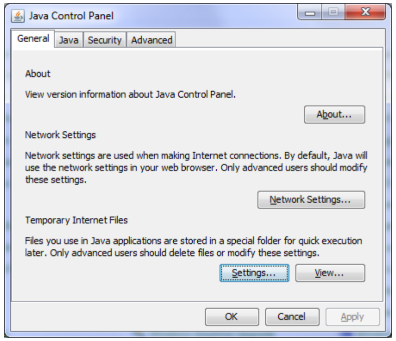
Instructor CCC Confer checklist

Universal Fix for CCC Confer:
1) Shrink (500 MB) and delete Java cache
2) Uninstall and reinstall latest Java runtime

Control Panel (small icons)



General Tab > Settings...



About

View version information about Java Control Panel.

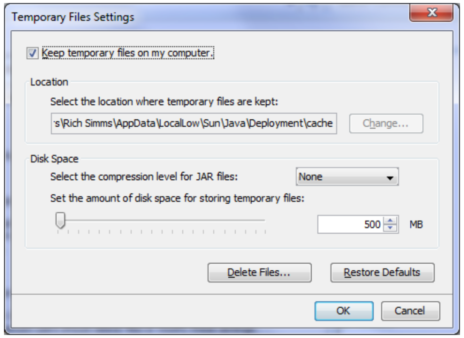
Network Settings

Network settings are used when making Internet connections. By default, Java will use the network settings in your web browser. Only advanced users should modify these settings.

Temporary Internet Files

Files you use in Java applications are stored in a special folder for quick execution later. Only advanced users should delete files or modify these settings.

500MB cache size



☑ Keep temporary files on my computer.

Location

Select the location where temporary files are kept:

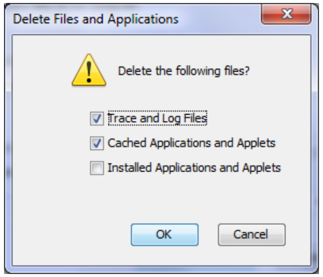
s\Rich Simms\AppData\LocalLow\Sun\Java\Deployment\cache

Disk Space

Select the compression level for JAR files: None

Set the amount of disk space for storing temporary files: 500 MB

Delete these



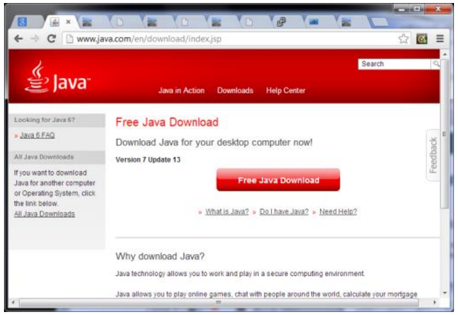
Delete the following files?

☑ Trace and Log Files

☑ Cached Applications and Applets

☐ Installed Applications and Applets

Google Java download



www.java.com/en/download/index.jsp

Looking for Java 6?

» Java 6 FAQ

All Java Downloads

If you want to download Java for another computer or Operating System, click the link below.

All Java Downloads

Free Java Download

Download Java for your desktop computer now!

Version 7 Update 13

» What is Java? » Do I have Java? » Need Help?

Why download Java?

Java technology allows you to work and play in a secure computing environment.

Java allows you to play online games, chat with people around the world, calculate your mortgage

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 • Programs/processes • Command line syntax • Environment variables • Metacharacters • Life of the shell • Docs • Wrap up

Questions

Questions

How this course works?

Past lesson material?

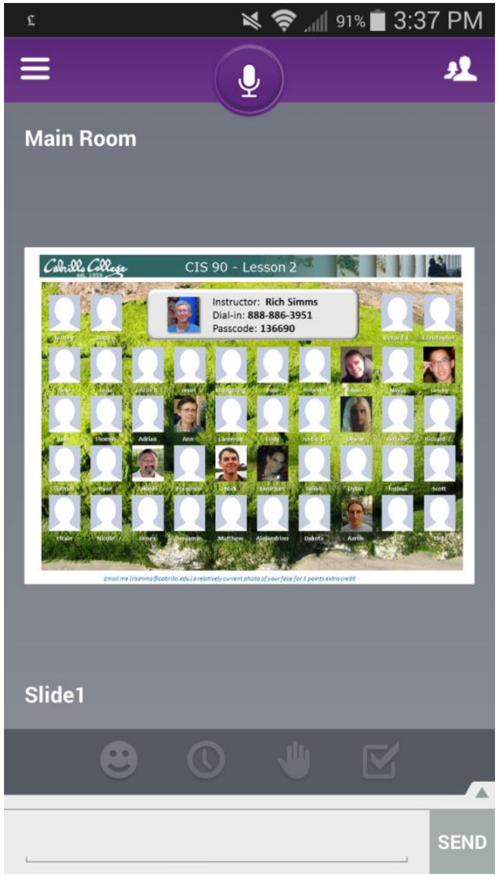
Previous labs?

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

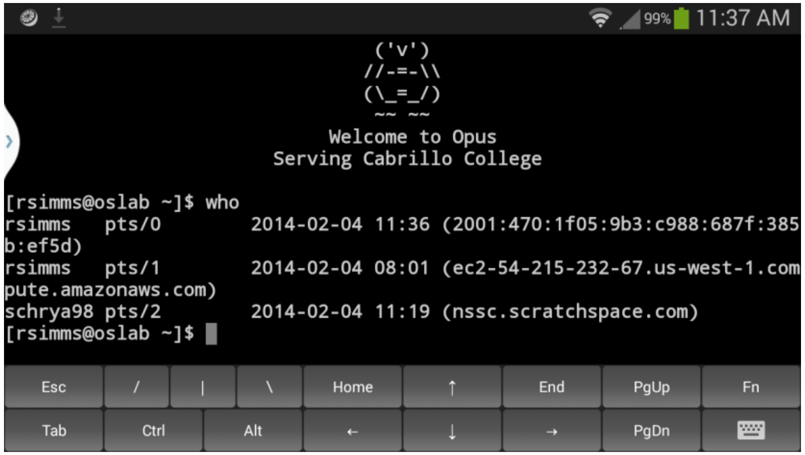
CIS 90 and Smartphones (Android)

Blackboard Collaborate App



Join CCC Confer virtual classroom

JuiceSSH - SSH Client app

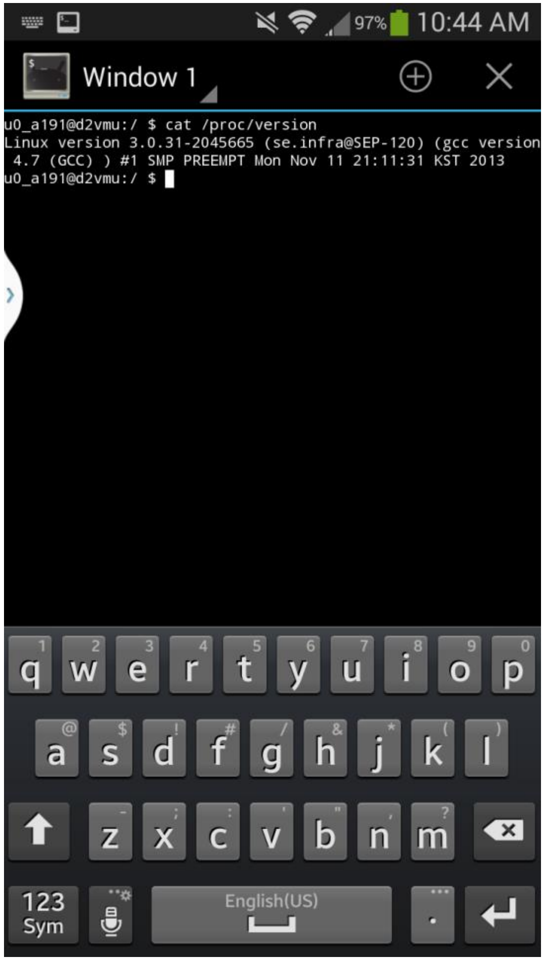


```
                        ('v')
                       //-=-\\
                       (\_=_/)
                        ~~ ~~
                    Welcome to Opus
                Serving Cabrillo College

[rsimms@oslab ~]$ who
rsimms   pts/0        2014-02-04 11:36 (2001:470:1f05:9b3:c988:687f:385
b:ef5d)
rsimms   pts/1        2014-02-04 08:01 (ec2-54-215-232-67.us-west-1.com
pute.amazonaws.com)
schrya98 pts/2        2014-02-04 11:19 (nssc.scratchspace.com)
[rsimms@oslab ~]$ 
```

Login to to Opus

CIS 90 and Smartphones (Android)

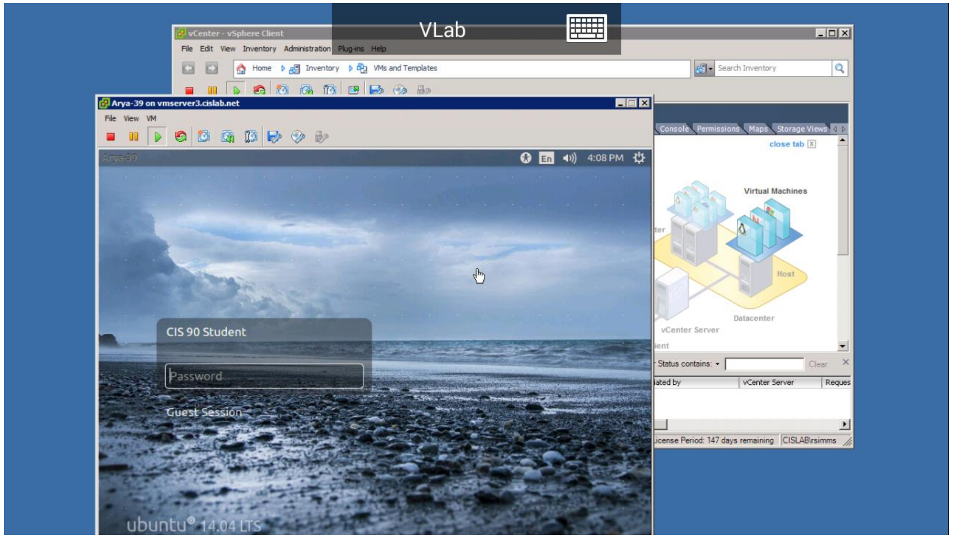
Android Terminal App



```
u0_a191@d2vmu:/ $ cat /proc/version
Linux version 3.0.31-2045665 (se.infra@SEP-120) (gcc version
 4.7 (GCC) ) #1 SMP PREEMPT Mon Nov 11 21:11:31 KST 2013
u0_a191@d2vmu:/ $
```

Viewing kernel version on smartphone

Microsoft RDP App



Running Arya VM in VLab

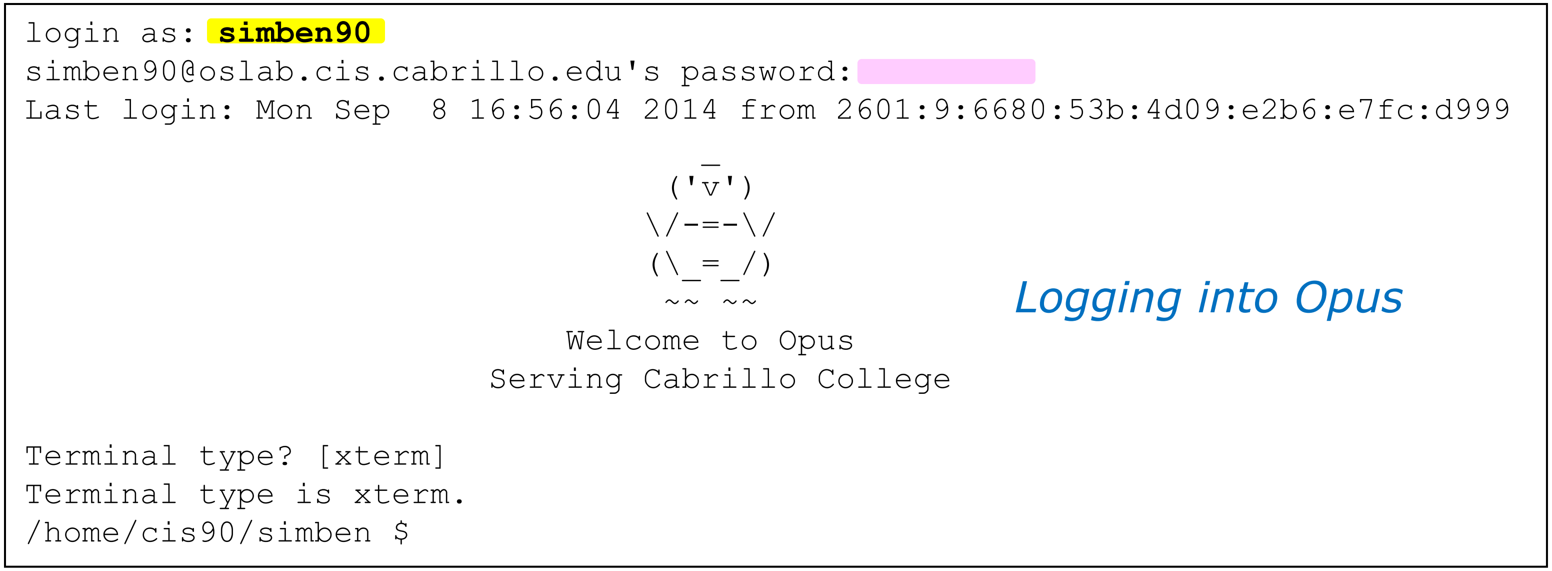
Logging In

Who goes there?

What's the password?

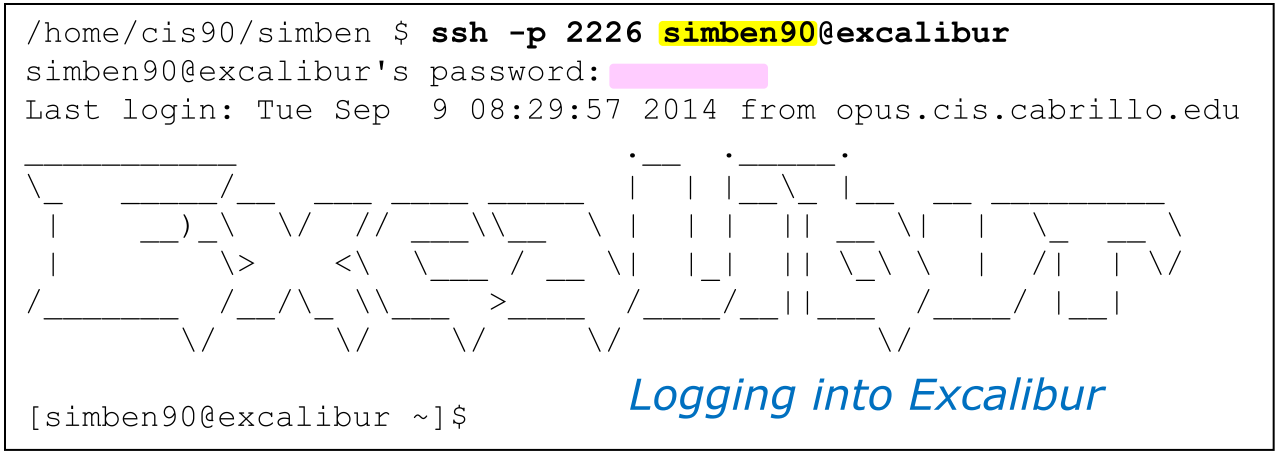
http://www.gutenberg.org/files/15064/15064-h/images/269.png

Logging In



```
login as: simben90
simben90@oslab.cis.cabrillo.edu's password:
Last login: Mon Sep  8 16:56:04 2014 from 2601:9:6680:53b:4d09:e2b6:e7fc:d999
                                   _
                                 ('v')
                                \/-=-\/
                                (\_=_/)
                                 ~~ ~~
                            Welcome to Opus
                        Serving Cabrillo College

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

Logging into Opus



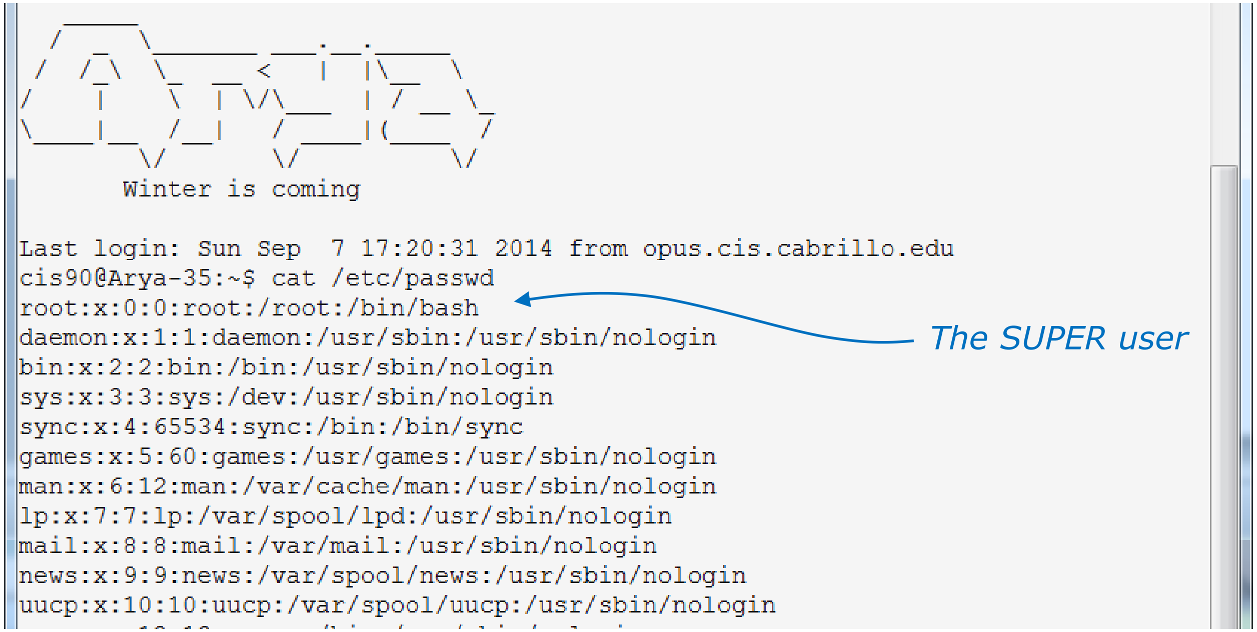
```
/home/cis90/simben $ ssh -p 2226 simben90@excalibur
simben90@excalibur's password:
Last login: Tue Sep  9 08:29:57 2014 from opus.cis.cabrillo.edu
___________                    .__  ._____.
\_   _____/__  ___ ____ _____  |  | |__\_ |__  __ _________
 |    __)_\  \/  // ___\\__  \ |  | |  || __ \|  |  \_  __ \
 |        \>    <\  \___ / __ \|  |_|  || \_\ \  |  /|  | \/
/_______  /__/\_ \\___  >____  /____/__||___  /____/ |__|
        \/      \/    \/     \/             \/

[simben90@excalibur ~]$
```

Logging into Excalibur

Don't expect to enter into a system without providing your username and password

/etc/passwd

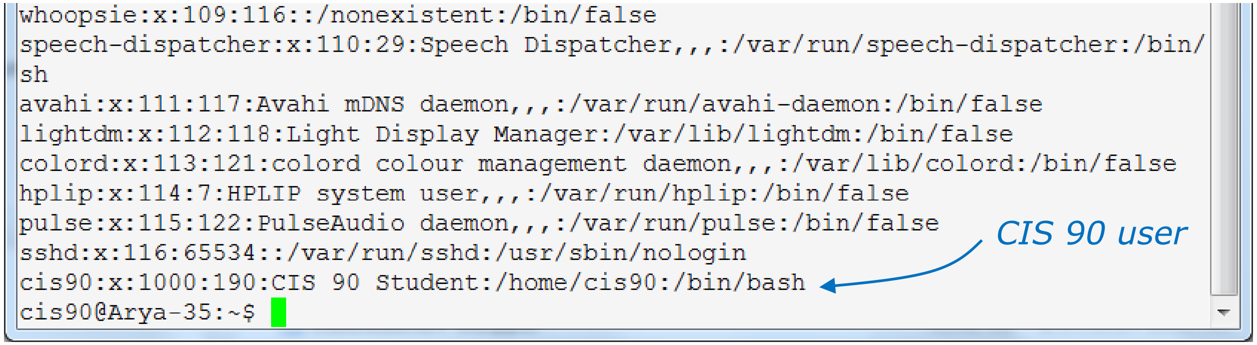
cat /etc/passwd



```
   _____
  /  _  \_______ ___.__._____
 /  /_\  \_  __ <   |  |\__  \
/    |    \  | \/\___  | / __ \_
\____|__  /__|   / ____|(____  /
        \/       \/          \/
       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
```

The SUPER user

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:~$
```

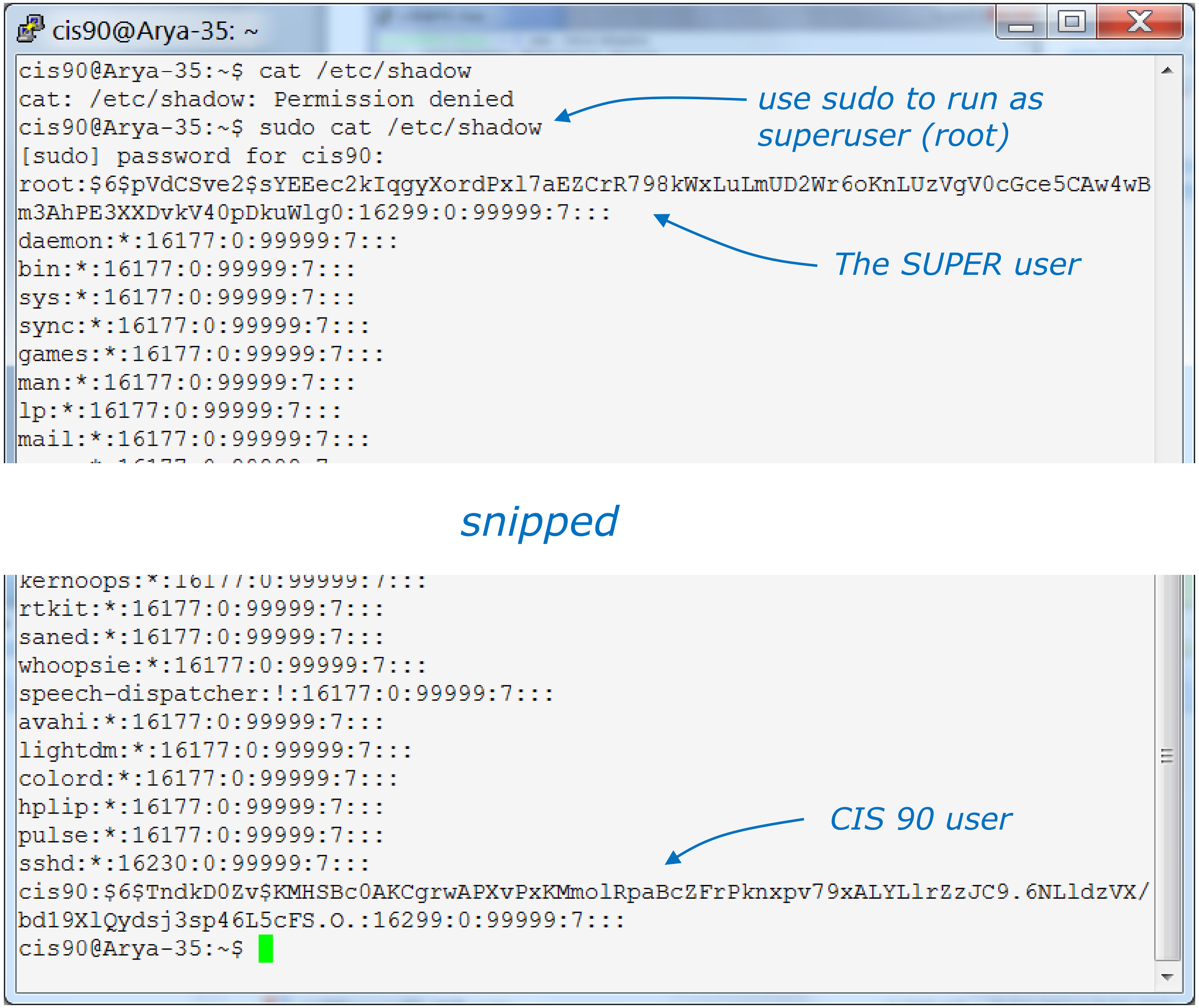
CIS 90 user

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

Note: this file no longer contains the passwords!

/etc/shadow

`cat /etc/shadow`



```
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$sYEEec2kIqgyXordPxl7aEZCrR798kWxLuLmUD2Wr6oKnLUzVgV0cGce5CAw4wB
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

snipped

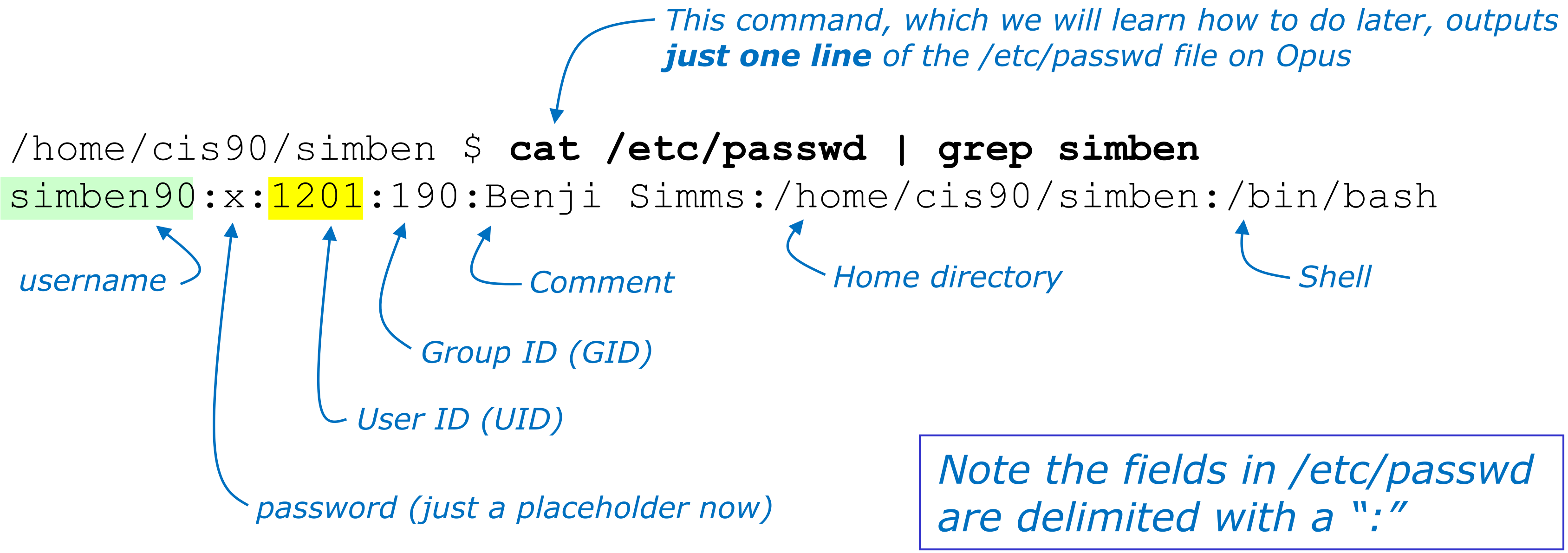
```
kernoops:*:16177:0:99999:7:::
rtkit:*:16177:0:99999:7:::
saned:*:16177:0:99999:7:::
whoopsie:*:16177:0:99999:7:::
speech-dispatcher:!:16177:0:99999:7:::
avahi:*:16177:0:99999:7:::
lightdm:*:16177:0:99999:7:::
colord:*:16177:0:99999:7:::
hplip:*:16177:0:99999:7:::
pulse:*:16177:0:99999:7:::
sshd:*:16230:0:99999:7:::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

CIS 90 user

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!

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

username

Comment

Home directory

Shell

Group ID (GID)

User ID (UID)

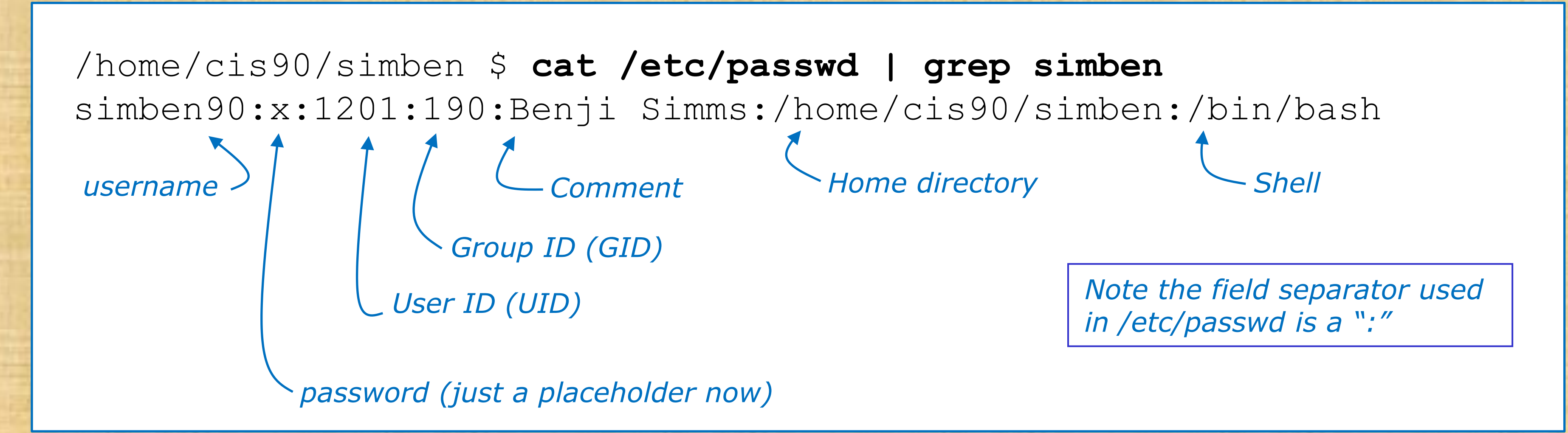
password (just a placeholder now)

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

username

password (just a placeholder now)

User ID (UID)

Group ID (GID)

Comment

Home directory

Shell

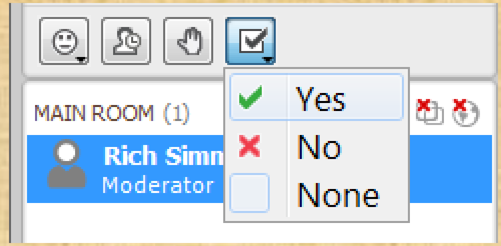
Note the field separator used in /etc/passwd is a ":"

1) Find your record in /etc/passwd

- Paste your User ID in the chat window
- Paste you home directory in the chat window
- Paste your shell in the chat window

2) cat /etc/shadow

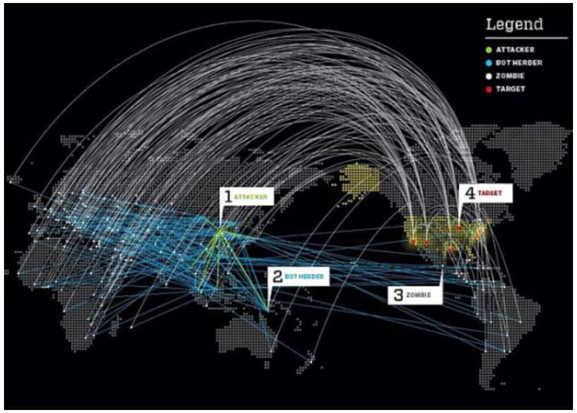
Give me a green check ✓ if you can view this file otherwise give me a red ✗



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://www.toptenreviews.com/i/rev/misc/articles/6956/botnet-zombie-a-1.jpg

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

They never stop trying

The firewall on Opus slows down but does not end the attacks

```
Failed logins from:
    122.249.183.95 (x183095.ppp.asahi-net.or.jp): 3 times
    218.64.5.131 (131.5.64.218.broad.nc.jx.dynamic.163data.com.cn): 3
times

 Illegal users from:
    78.46.83.76 (static.76.83.46.78.clients.your-server.de): 3 times
    218.4.157.178: 3 times

 pam_succeed_if(sshd:auth): error retrieving information about user
teamspeak : 1 time(s)
 reverse mapping checking getaddrinfo for
131.5.64.218.broad.nc.jx.dynamic.163data.com.cn failed - POSSIBLE
BREAK-IN ATTEMPT! : 3 time(s)
 pam_succeed_if(sshd:auth): error retrieving information about user ts
: 2 time(s)
 pam_succeed_if(sshd:auth): error retrieving information about user
plcmspip : 2 time(s)
 pam_succeed_if(sshd:auth): error retrieving information about user
PlcmSpIp : 1 time(s)
```

We used to get up thousands of attempts every day until we made some changes to the firewall on Opus. Attacks always would come from different computers around the world.

/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]#
```

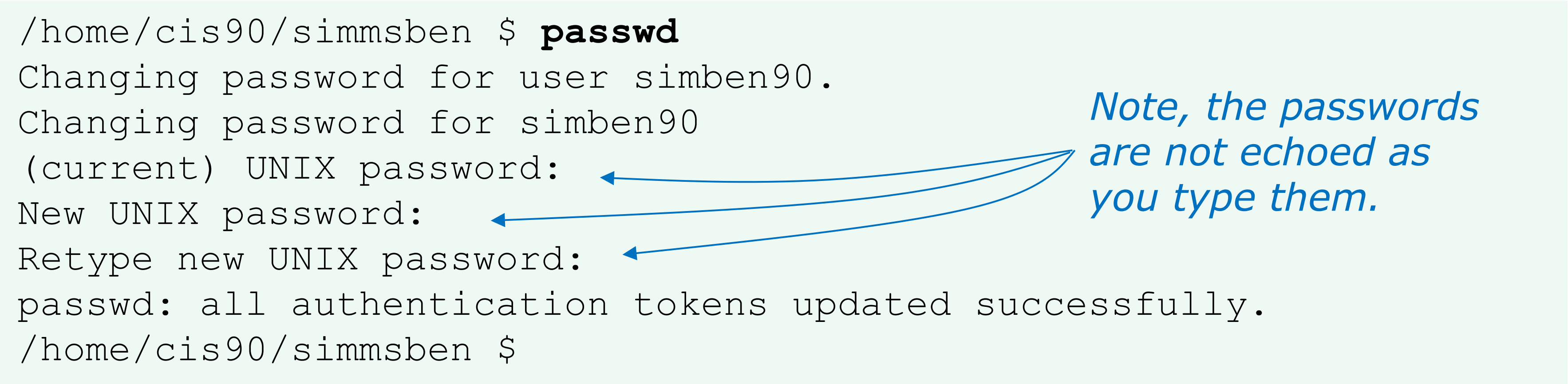
Top 50 usernames used by the ne'er-do-wells

How to make a strong password

- 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)

How to change your password on Opus



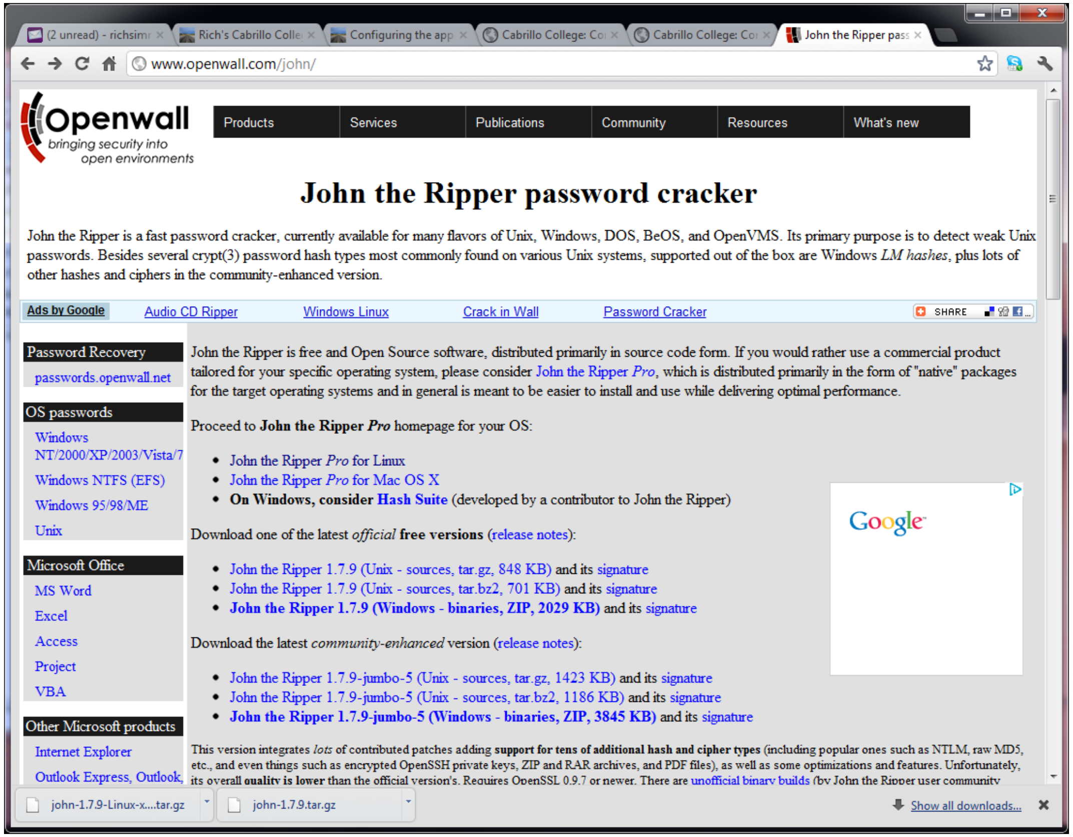
```
/home/cis90/simmsben $ passwd
Changing password for user simben90.
Changing password for simben90
(current) UNIX password:
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
/home/cis90/simmsben $
```

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

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

John the Ripper

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



John the Ripper password cracker

John the Ripper is a fast password cracker, currently available for many flavors of Unix, Windows, DOS, BeOS, and OpenVMS. Its primary purpose is to detect weak Unix passwords. Besides several crypt(3) password hash types most commonly found on various Unix systems, supported out of the box are Windows *LM hashes*, plus lots of other hashes and ciphers in the community-enhanced version.

Password Recovery

- passwords.openwall.net

OS passwords

- Windows NT/2000/XP/2003/Vista/7
- Windows NTFS (EFS)
- Windows 95/98/ME
- Unix

Microsoft Office

- MS Word
- Excel
- Access
- Project
- VBA

Other Microsoft products

- Internet Explorer
- Outlook Express, Outlook,

John the Ripper is free and Open Source software, distributed primarily in source code form. If you would rather use a commercial product tailored for your specific operating system, please consider John the Ripper *Pro*, which is distributed primarily in the form of "native" packages for the target operating systems and in general is meant to be easier to install and use while delivering optimal performance.

Proceed to **John the Ripper *Pro*** homepage for your OS:

- John the Ripper *Pro* for Linux
- John the Ripper *Pro* for Mac OS X
- **On Windows, consider Hash Suite** (developed by a contributor to John the Ripper)

Download one of the latest *official* **free versions** (release notes):

- John the Ripper 1.7.9 (Unix - sources, tar.gz, 848 KB) and its signature
- John the Ripper 1.7.9 (Unix - sources, tar.bz2, 701 KB) and its signature
- **John the Ripper 1.7.9 (Windows - binaries, ZIP, 2029 KB)** and its signature

Download the latest *community-enhanced* version (release notes):

- John the Ripper 1.7.9-jumbo-5 (Unix - sources, tar.gz, 1423 KB) and its signature
- John the Ripper 1.7.9-jumbo-5 (Unix - sources, tar.bz2, 1186 KB) and its signature
- **John the Ripper 1.7.9-jumbo-5 (Windows - binaries, ZIP, 3845 KB)** and its signature

This version integrates *lots* of contributed patches adding **support for tens of additional hash and cipher types** (including popular ones such as NTLM, raw MD5, etc., and even things such as encrypted OpenSSH private keys, ZIP and RAR archives, and PDF files), as well as some optimizations and features. Unfortunately, its overall **quality is lower** than the official version's. Requires OpenSSL 0.9.7 or newer. There are unofficial binary builds (by John the Ripper user community

john-1.7.9/run/password.lst has most popular passwords to try first

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 your turned in

3. Last day to drop/add is Saturday 9/13

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

simms-teach.com/docs/cis90/cis90lab01.p

CIS 90 UNIX/Linux Lab Exercise

Lab 1: Scavenger hunt

Fall 2014

Lab 1: Scavenger hunt

In this lab you go on an electronic scavenger hunt to collect various items from different systems. These systems can be accessed remotely via a network connection. You will use the ssh command to login into each system and then use the Lesson 1 commands to answer each question. Once you correctly answer all the questions you will receive the scavenger hunt item for that system.

UNIX/Linux Systems

System name	Scavenger hunt item to collect
Defiant	A star
Enterprise	A movie
Excalibur	A dog breed
Intrepid	A fruit
Lexington	A musical instrument
Oslab (Opus)	No items to collect here. Start and end the scavenger hunt on Opus.

Systems Roadmap

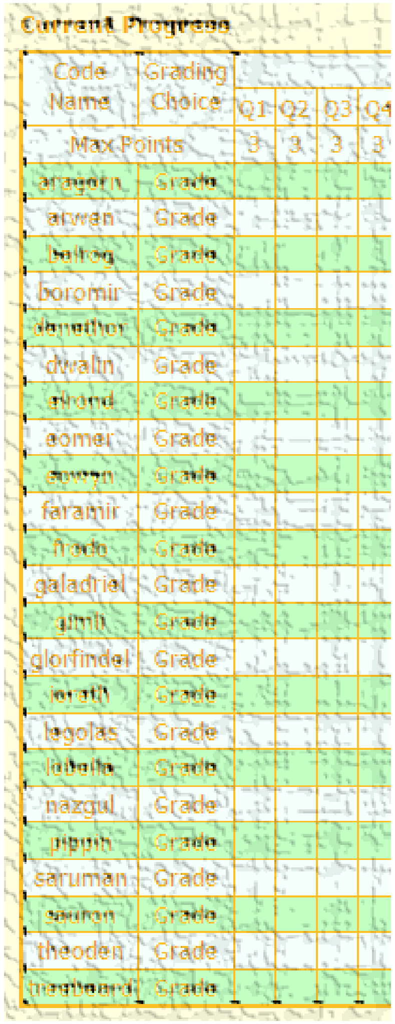
Pearls of Wisdom:

- 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

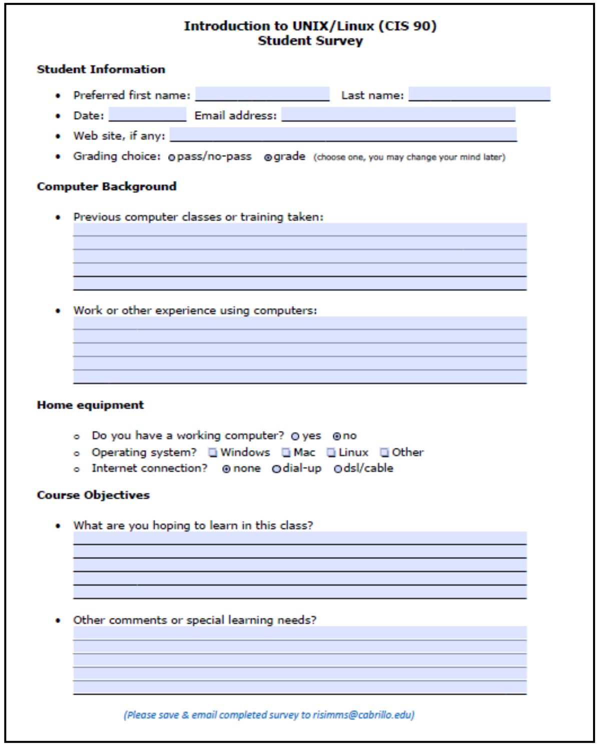
Lord of the Rings Characters

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



Current Progress

Code Name	Grading Choice	Q1	Q2	Q3	Q4
Max Points		3	3	3	3
aragorn	Grade				
arwen	Grade				
balrog	Grade				
boromir	Grade				
denethor	Grade				
dwalin	Grade				
elrond	Grade				
eomer	Grade				
eowyn	Grade				
faramir	Grade				
frodo	Grade				
galadriel	Grade				
gimli	Grade				
glorfindel	Grade				
ioreth	Grade				
legolas	Grade				
lobelia	Grade				
nazgul	Grade				
pippin	Grade				
saruman	Grade				
sauron	Grade				
theoden	Grade				
treebeard	Grade				



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?

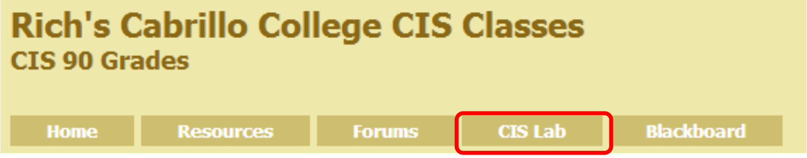
(Please save & email completed survey to risimms@cabrillo.edu)

Help Available in the CIS Lab

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

The CIS Lab

Inside the MESA Center

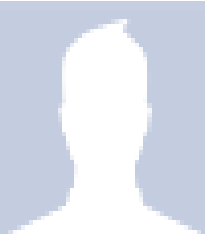


Rich's Cabrillo College CIS Classes

CIS 90 Grades

CIS 90 Lab Assistants:

Geoff



Leandro

Mo-Th 8-11am
(except Th Sept 11)
Fr 9:30-12:30

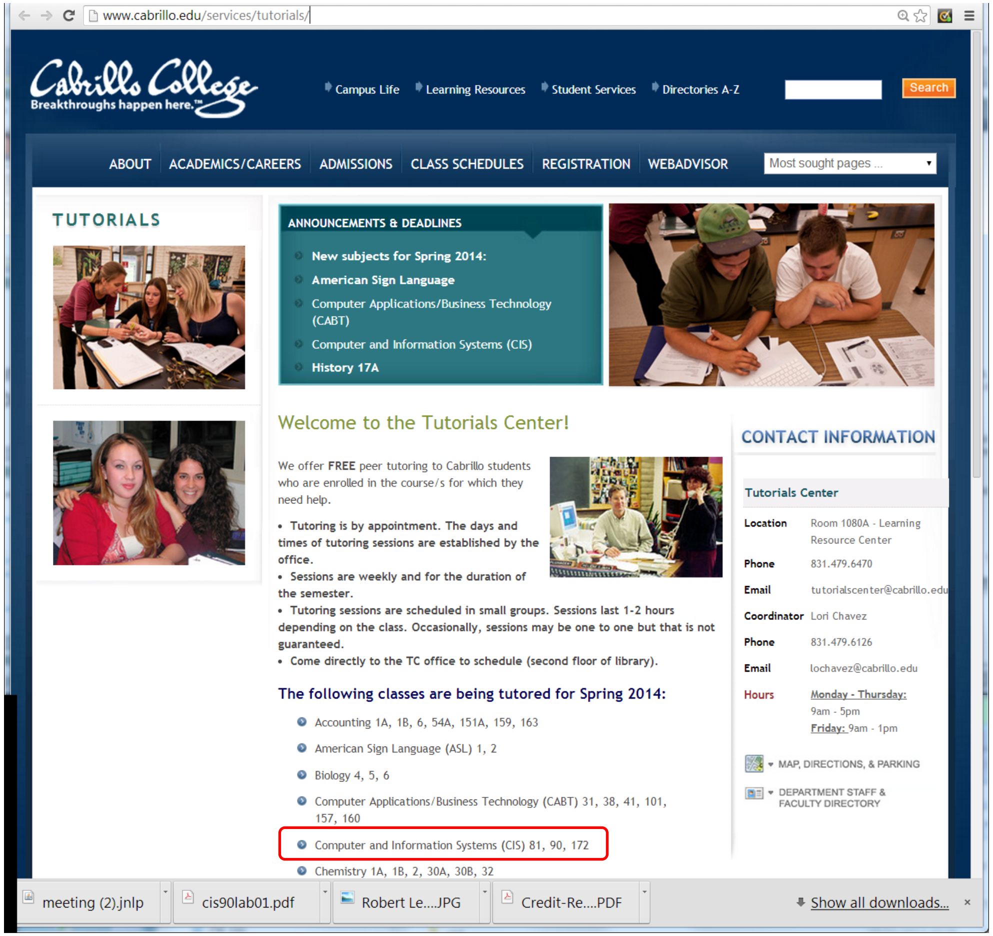
Linux Instructors

Michael Matera

Look for Geoff, Leandro or Michael on the schedule found here

CIS 90 Tutoring Available

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

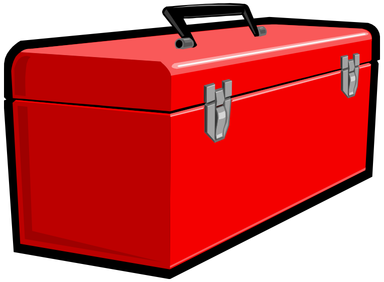
Location	Room 1080A - Learning Resource Center
Phone	831.479.6470
Email	tutorialscenter@cabrillo.edu
Coordinator	Lori Chavez
Phone	831.479.6126
Email	lochavez@cabrillo.edu
Hours	Monday - Thursday: 9am - 5pm Friday: 9am - 1pm

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	*Print text and variables*
banner	*Make a banner*

ls	*List directory contents*
cat	*View file (name comes from concatenate)*
file	*Show additional information about a file*
type	*Shows where a command resides on the path*
apropos	*Searches the whatis database for strings*
whatis	*Searches the whatis database for commands*
man	*Show the manual page for a command*
info	*Alternate online documentation tool*

bc	*Binary calculator*
passwd	*Change password*

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 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
```

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
```

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

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)
```

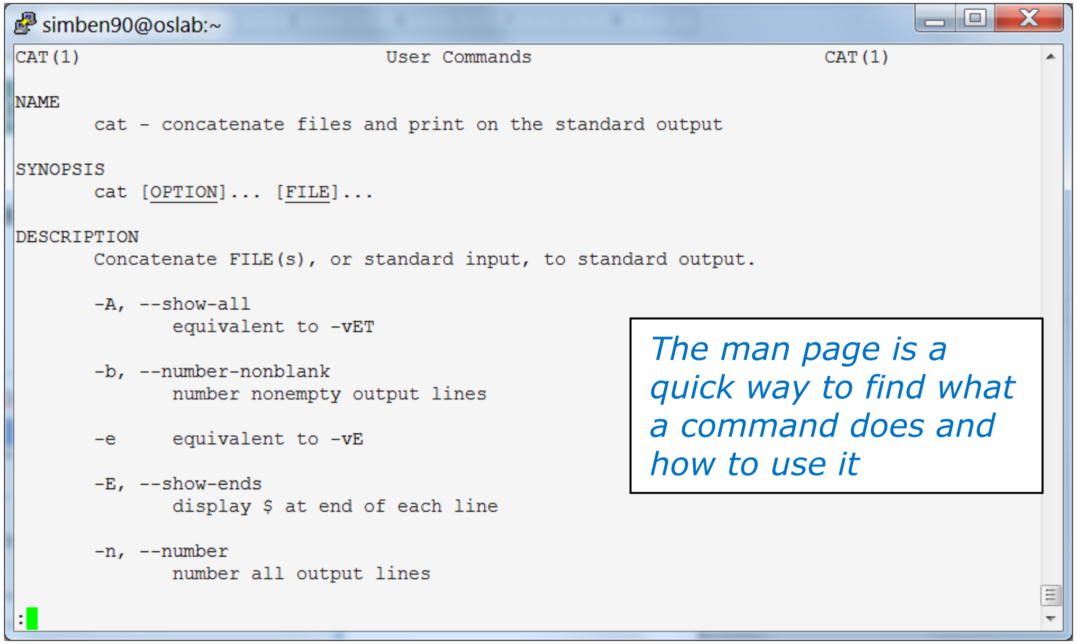
man command

Show the manual page (documentation) for a command

Syntax:

man *command*

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



```
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

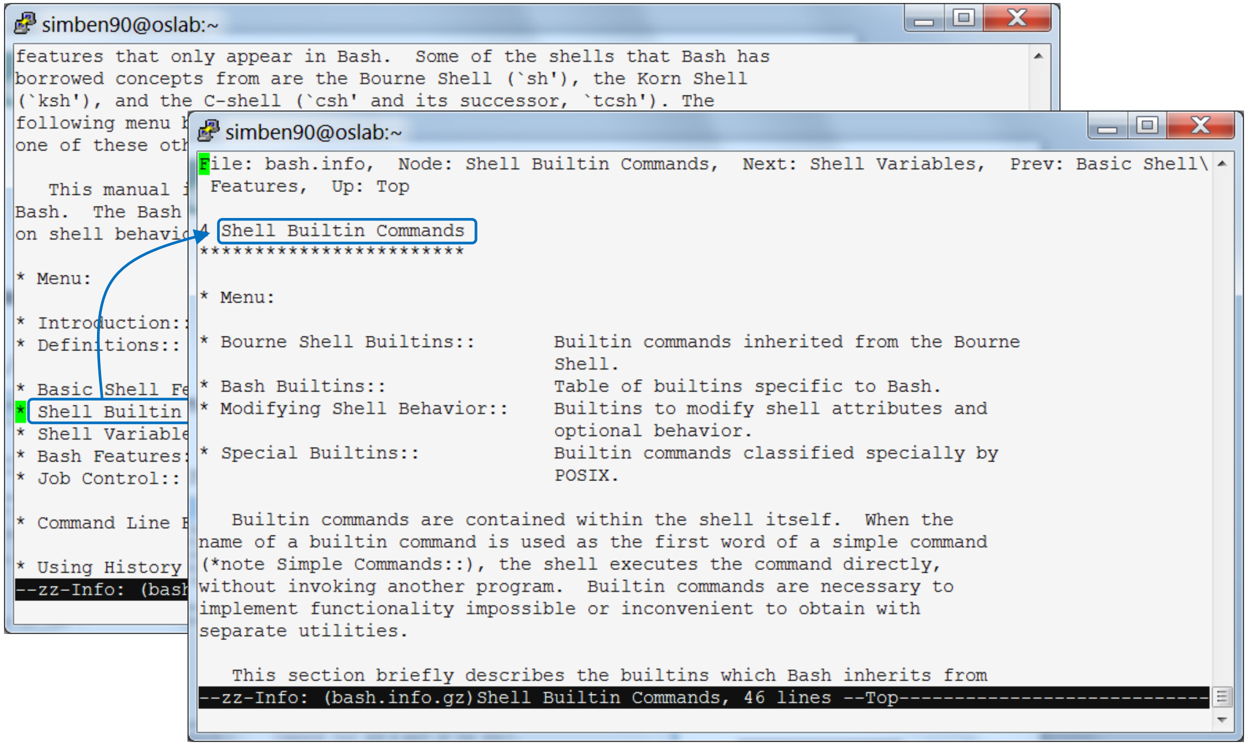
Alternate documentation tool for commands

Syntax:

info *command*

Similar to man but has has links to additional pages

```
/home/cis90/simben $ info echo
```



```
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 b
one of these oth

   This manual i
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: (bash
```

```
File: bash.info,  Node: Shell Builtin Commands,  Next: Shell Variables,  Prev: Basic Shell\
 Features,  Up: Top

4 Shell Builtin Commands
************************

* Menu:

* Bourne Shell Builtins::       Builtin commands inherited from the Bourne
                                Shell.
* Bash Builtins::               Table of builtins specific to Bash.
* Modifying Shell Behavior::    Builtins to modify shell attributes and
                                optional behavior.
* Special Builtins::            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*

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 equations for bc to solve

Use quit to end program

Class Activity

1) Where is the **bc** command?
   Type your answer in the chat window.

2) Is the **bc** command a binary executable or a shell script?
   Type your answer in the chat window.

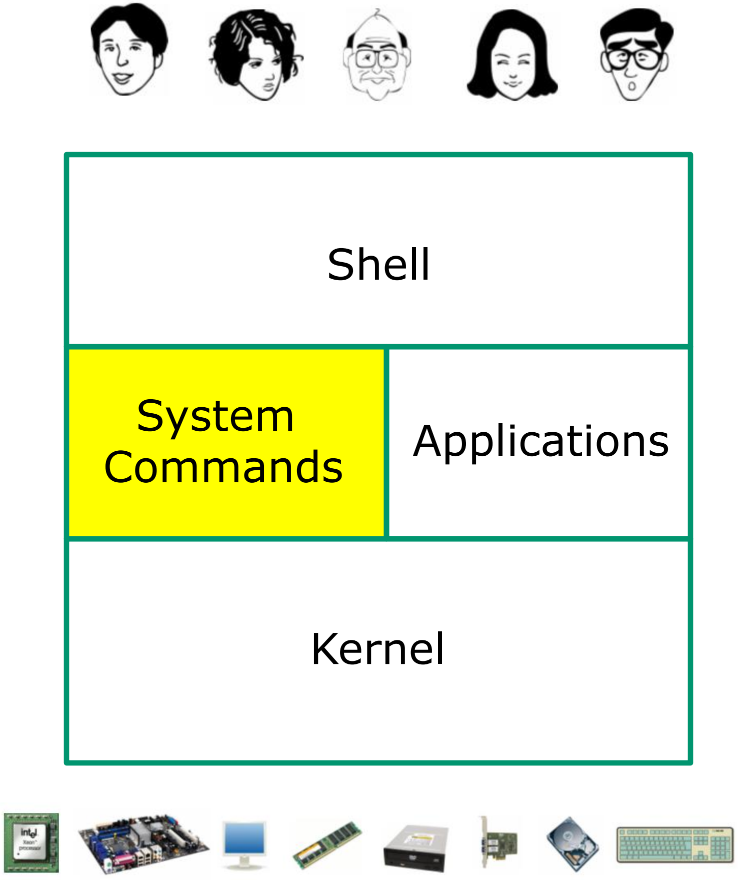
3) Can you **cat** the **bc** command?
   Paste a line of output in the chat window.

4) Is **bc** a UNIX command? Hint: use the **man** or **whatis** commands with bc as the argument.
   Type your answer in the chat window.

Where are the UNIX/Linux commands?

UNIX/Linux Architecture

System Commands

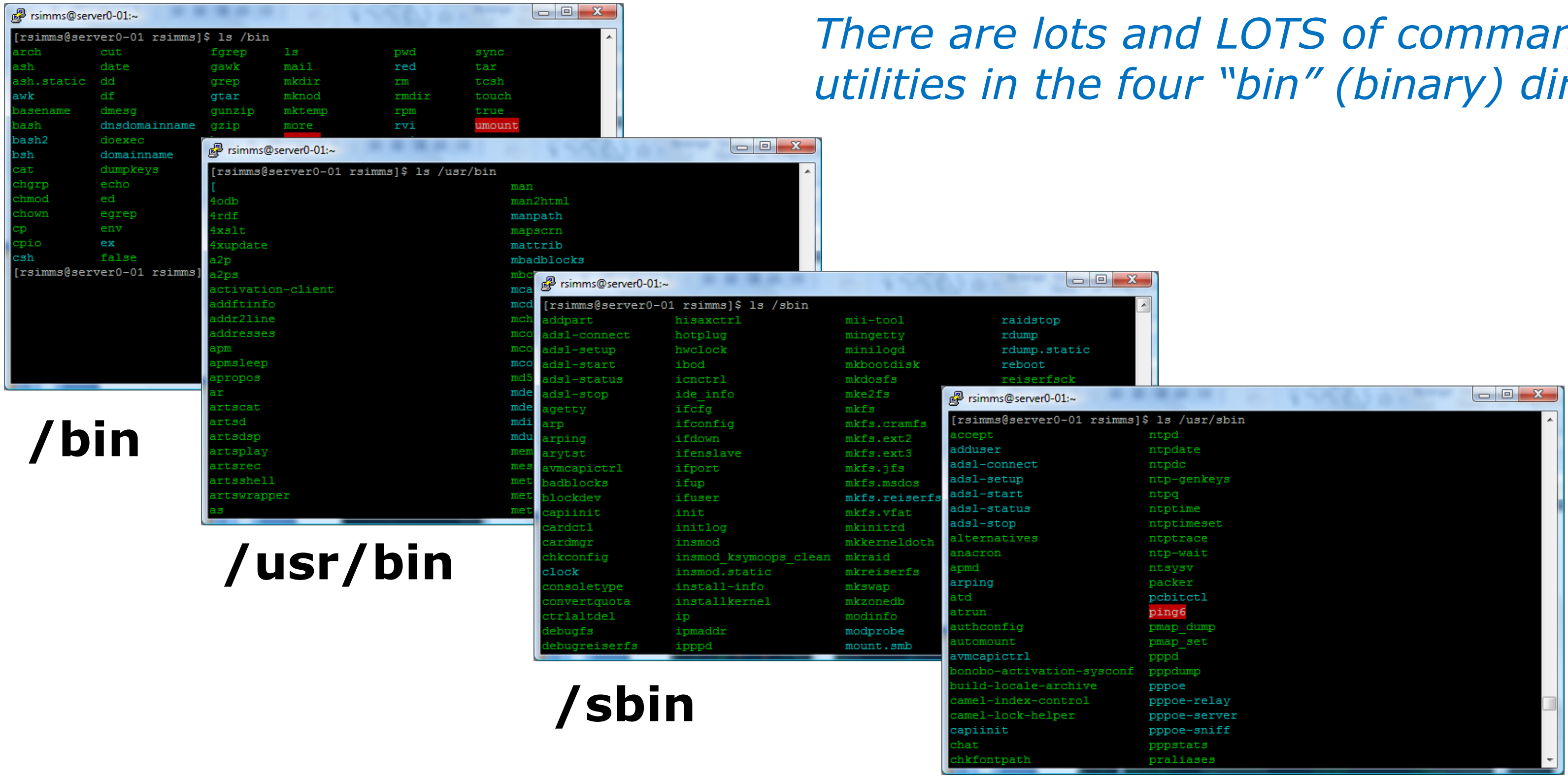


- 100’s of system commands and utilities .
- Commands like **ls** (list directories), **cat** (print a file), **rm** (remove a file), … etc.
- Utilities like **vi** (text editor), **sort** (sorts file contents), **find** (searches), … etc.
- Larger utilities like **sendmail** (email), **tar** (backup),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

Commands and Utilities

Executable binary code (programs) or scripts

There are lots and LOTS of commands & utilities in the four "bin" (binary) directories



```
[rsimms@server0-01 rsimms]$ ls /bin
arch        cut            fgrep     ls         pwd      sync
ash         date           gawk      mail       red      tar
ash.static  dd             grep      mkdir      rm       tcsh
awk         df             gtar      mknod      rmdir    touch
basename    dmesg          gunzip    mktemp     rpm      true
bash        dnsdomainname  gzip      more       rvi      umount
bash2       doexec
bsh         domainname
cat         dumpkeys
chgrp       echo
chmod       ed
chown       egrep
cp          env
cpio        ex
csh         false
[rsimms@server0-01 rsimms]
```

/bin

```
[rsimms@server0-01 rsimms]$ ls /usr/bin
[                                           man
4odb                                        man2html
4rdf                                        manpath
4xslt                                       mapscrn
4xupdate                                    mattrib
a2p                                         mbadblocks
a2ps                                        mbc
activation-client                           mca
addftinfo                                   mcd
addr2line                                   mch
addresses                                   mco
apm                                         mco
apmsleep                                    mco
apropos                                     md5
ar                                          mde
artscat                                     mde
artsd                                       mdi
artsdsp                                     mdu
artsplay                                    mem
artsrec                                     mes
artsshell                                   met
artswrapper                                 met
as                                          met
```

/usr/bin

```
[rsimms@server0-01 rsimms]$ ls /sbin
addpart          hisaxctrl                mii-tool          raidstop
adsl-connect     hotplug                  mingetty          rdump
adsl-setup       hwclock                  minilogd          rdump.static
adsl-start       ibod                     mkbootdisk        reboot
adsl-status      icnctrl                  mkdosfs           reiserfsck
adsl-stop        ide_info                 mke2fs
agetty           ifcfg                    mkfs
arp              ifconfig                 mkfs.cramfs
arping           ifdown                   mkfs.ext2
arytst           ifenslave                mkfs.ext3
avmcapictrl      ifport                   mkfs.jfs
badblocks        ifup                     mkfs.msdos
blockdev         ifuser                   mkfs.reiserfs
capiinit         init                     mkfs.vfat
cardctl          initlog                  mkinitrd
cardmgr          insmod                   mkkerneldoth
chkconfig        insmod_ksymoops_clean    mkraid
clock            insmod.static            mkreiserfs
consoletype      install-info             mkswap
convertquota     installkernel            mkzonedb
ctrlaltdel       ip                       modinfo
debugfs          ipmaddr                  modprobe
debugreiserfs    ipppd                    mount.smb
```

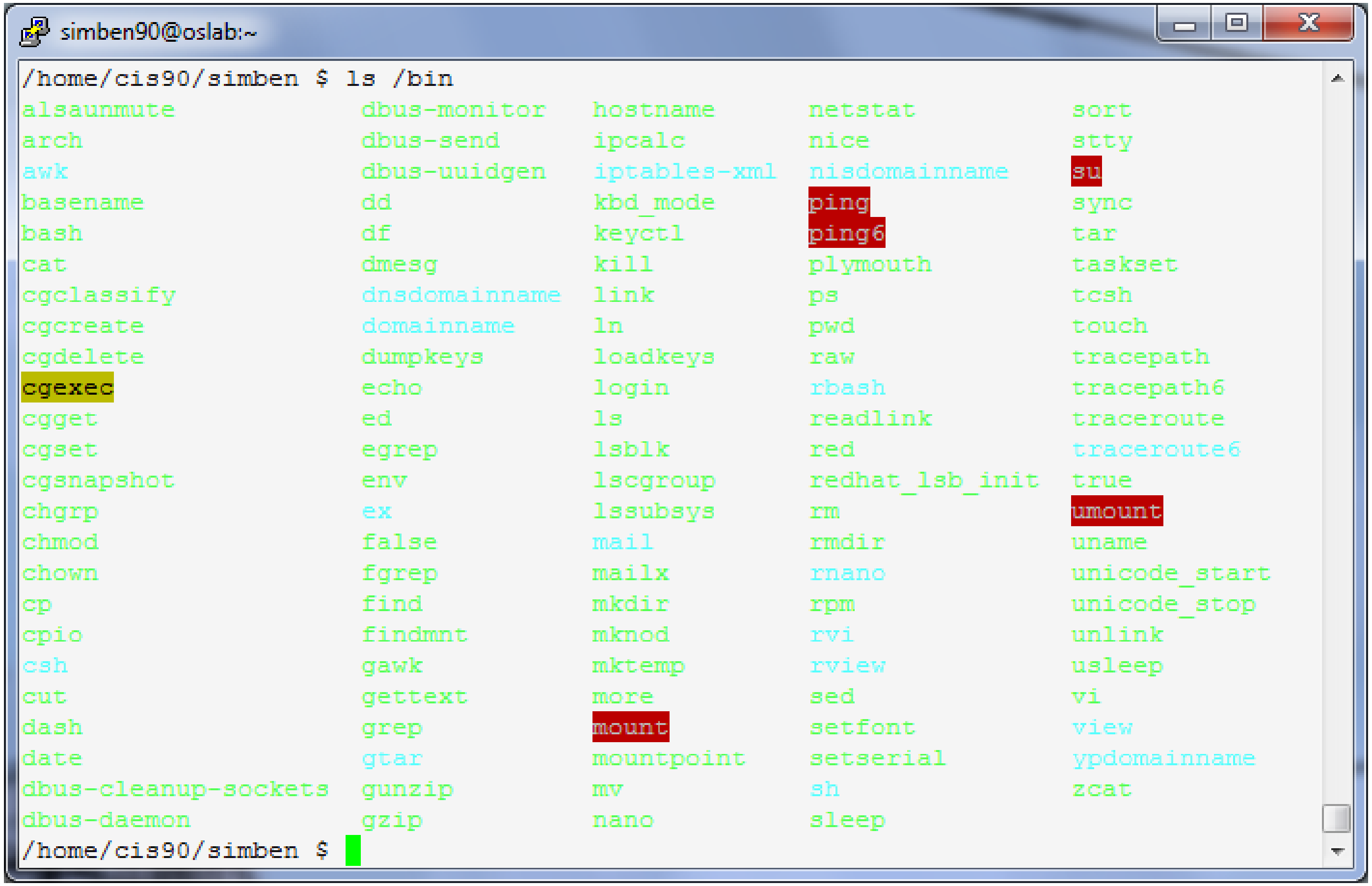
/sbin

```
[rsimms@server0-01 rsimms]$ ls /usr/sbin
accept                       ntpd
adduser                      ntpdate
adsl-connect                 ntpdc
adsl-setup                   ntp-genkeys
adsl-start                   ntpq
adsl-status                  ntptime
adsl-stop                    ntptimeset
alternatives                 ntptrace
anacron                      ntp-wait
apmd                         ntsysv
arping                       packer
atd                          pcbitctl
atrun                        ping6
authconfig                   pmap_dump
automount                    pmap_set
avmcapictrl                  pppd
bonobo-activation-sysconf    pppdump
build-locale-archive         pppoe
camel-index-control          pppoe-relay
camel-lock-helper            pppoe-server
capiinit                     pppoe-sniff
chat                         pppstats
chkfontpath                  praliases
```

/usr/sbin

The /bin directory

ls /bin



```
/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-cleanup-sockets  gunzip         mv            sh               zcat
dbus-daemon           gzip           nano          sleep
/home/cis90/simben $
```

/bin has essential commands used by everyone.

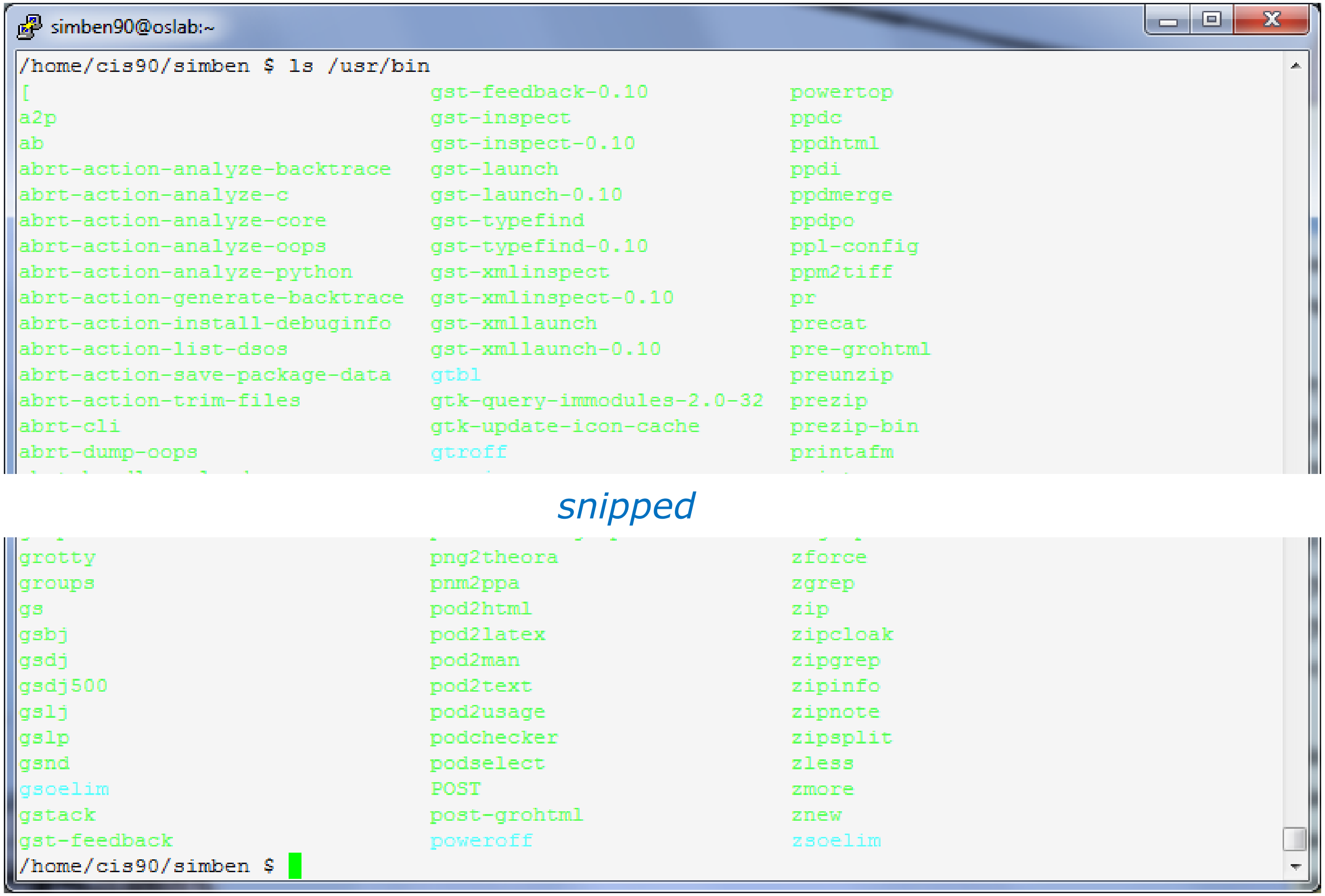
Can you find the Lesson 1 ***date****,* ***hostname****,* ***ps*** *and* ***uname*** *commands?*

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

Commands are either program or script files that can be executed

The /usr/bin directory

ls /usr/bin



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

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
gsoelim                         POST                        zmore
gstack                          post-grohtml                znew
gst-feedback                    poweroff                    zsoelim
/home/cis90/simben $
```

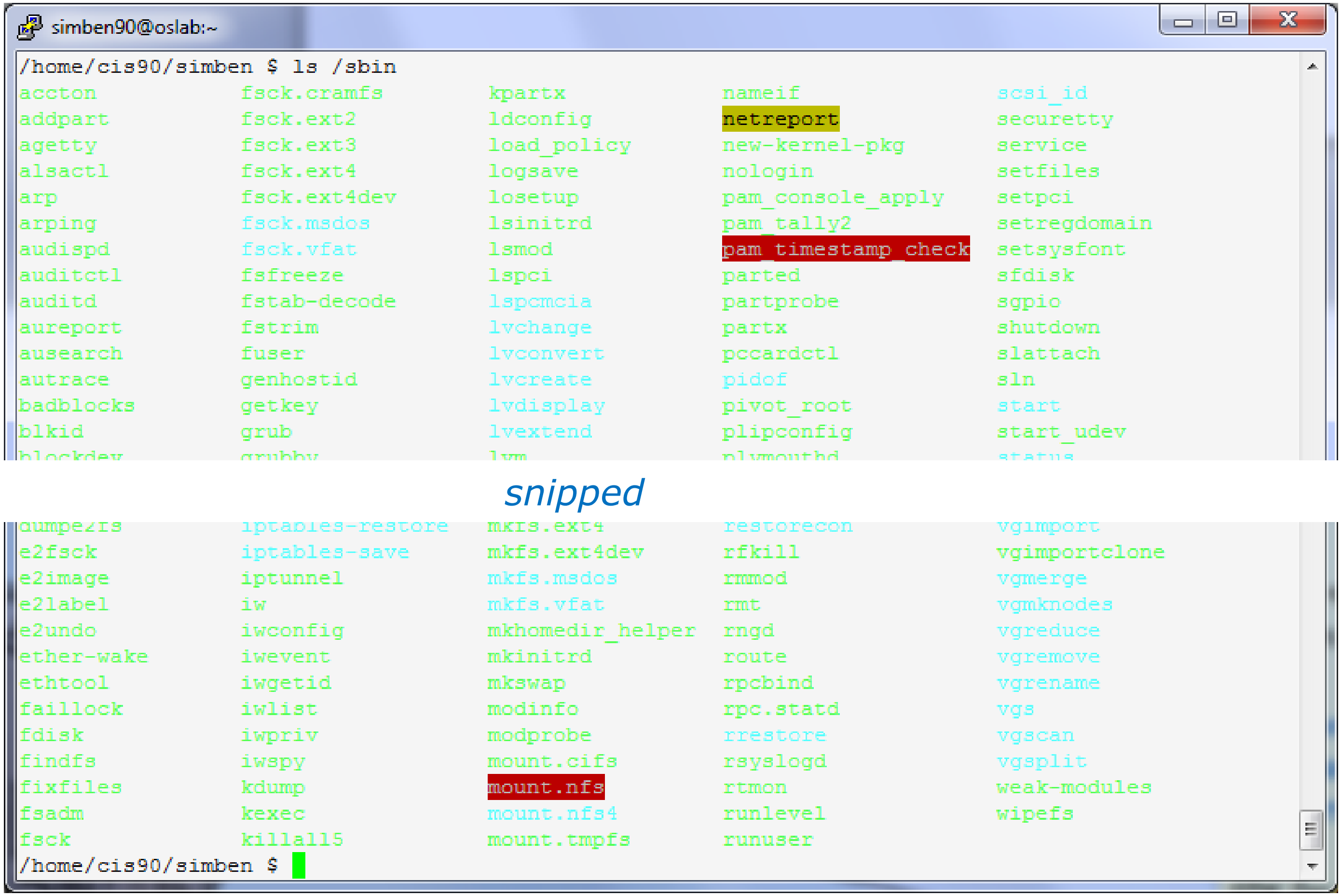
There are a "ton" of additional commands (programs) in this directory.

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

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              setfiles
arp              fsck.ext4dev       losetup           pam_console_apply    setpci
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             lvm               plymouthd            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                vgremove
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 $
```

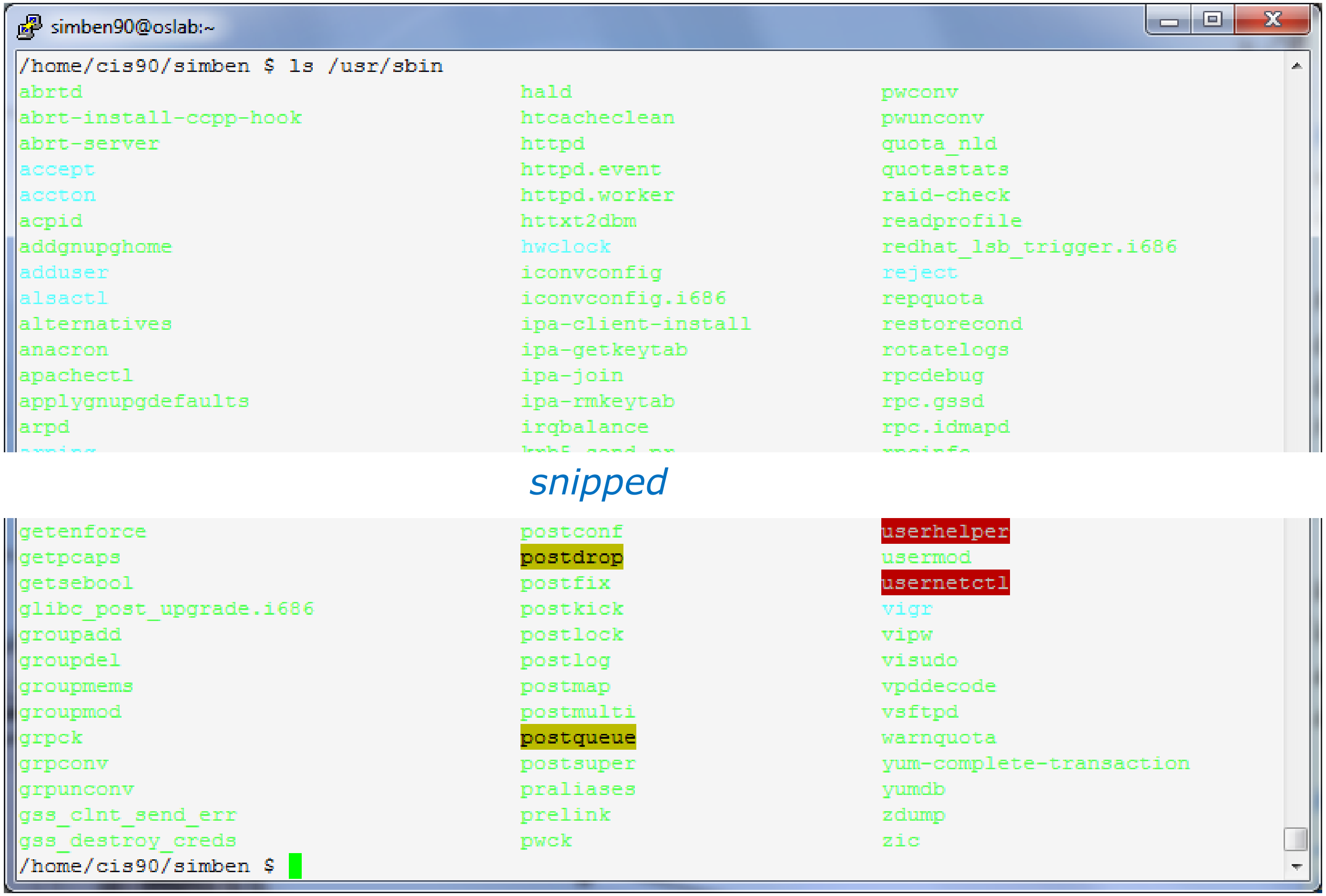
These are essential commands and utilities used by system administrators.

*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`



```
/home/cis90/simben $ ls /usr/sbin
abrtd                                  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               rotatelogs
apachectl                              ipa-join                    rpcdebug
applygnupgdefaults                     ipa-rmkeytab                rpc.gssd
arpd                                   irqbalance                  rpc.idmapd
arping                                 krb5_send_rr                rpcinfo
```

snipped

```
getenforce                             postconf                    userhelper
getpcaps                               postdrop                    usermod
getsebool                              postfix                     usernetctl
glibc_post_upgrade.i686                postkick                    vigr
groupadd                               postlock                    vipw
groupdel                               postlog                     visudo
groupmems                              postmap                     vpddecode
groupmod                               postmulti                   vsftpd
grpck                                  postqueue                   warnquota
grpconv                                postsuper                   yum-complete-transaction
grpunconv                              praliases                   yumdb
gss_clnt_send_err                      prelink                     zdump
gss_destroy_creds                      pwck                        zic
/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.

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 binary code.
- 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
```

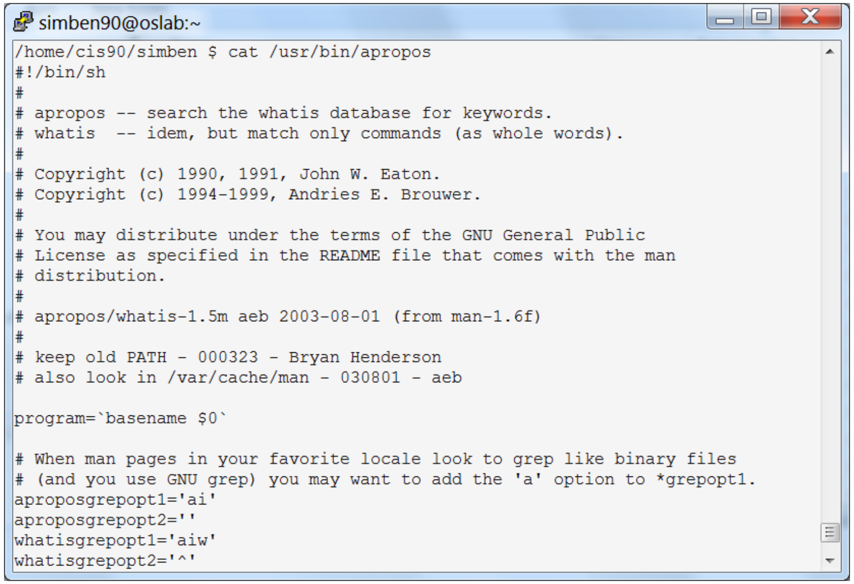
***apropos** is a shell script*

```
/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

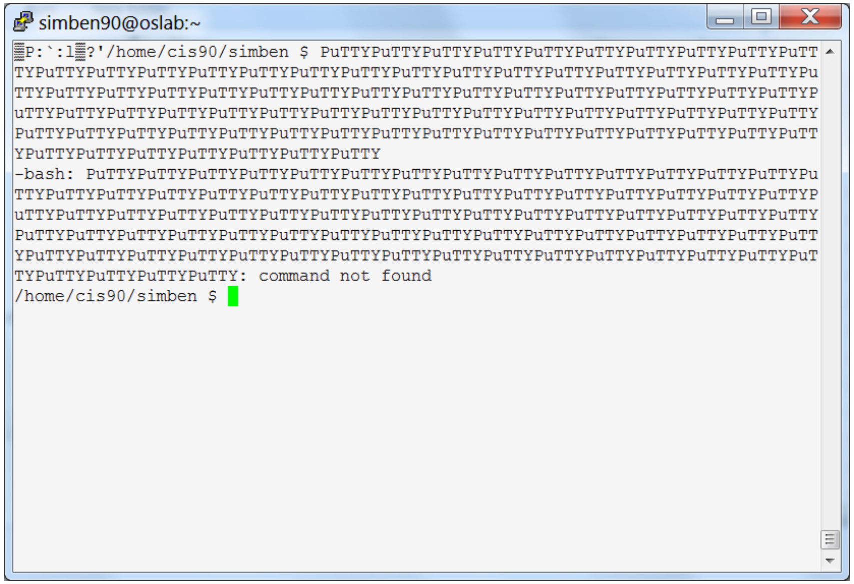
`cat /usr/bin/apropos`



```
/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.
aproposgrepopt1='ai'
aproposgrepopt2=''
whatisgrepopt1='aiw'
whatisgrepopt2='^'
```

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

`cat /usr/bin/cal`



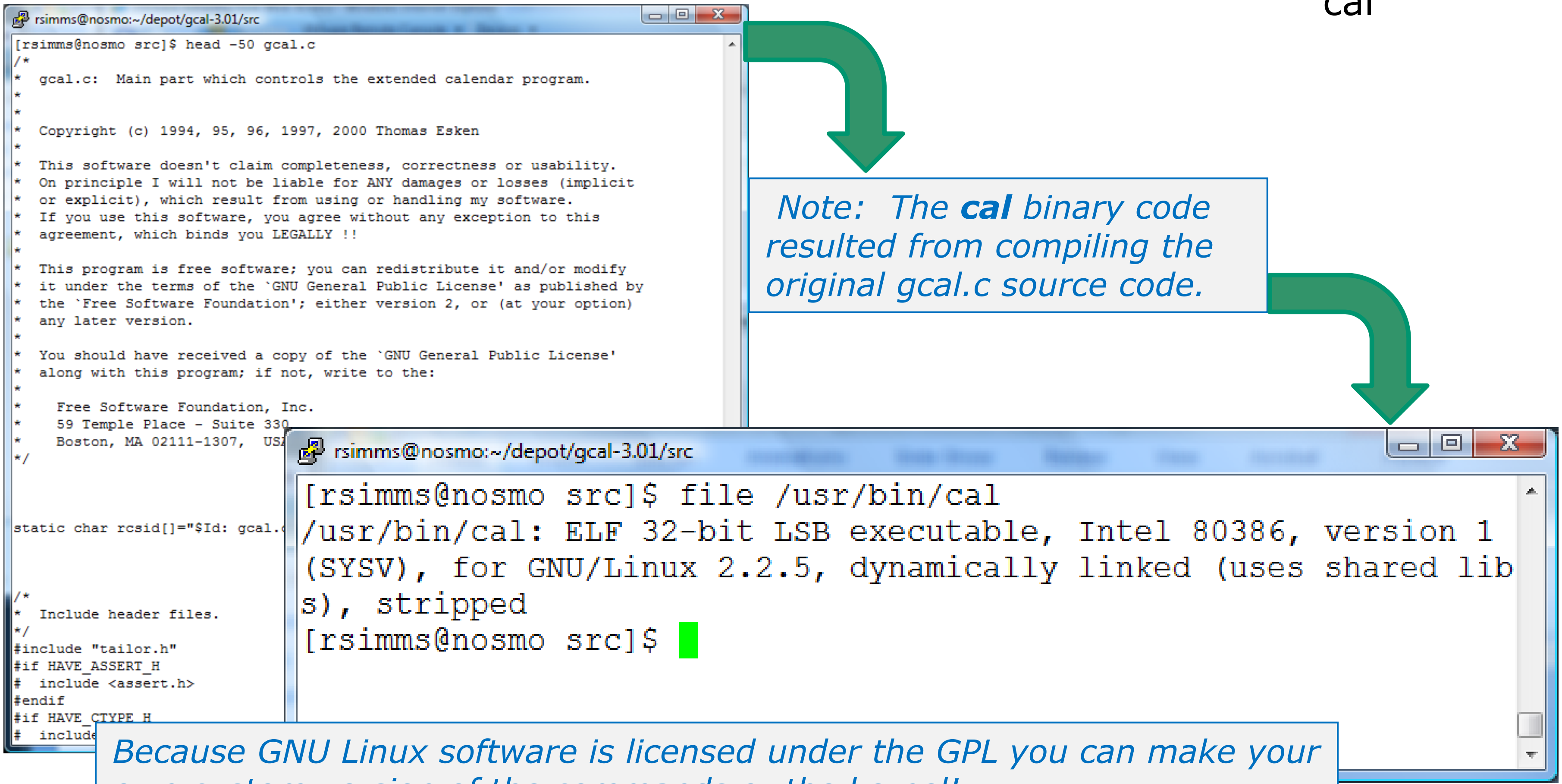
```
▒P:`:l▒?'/home/cis90/simben $ PuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTT
TYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPu
TTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYP
uTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTY
PuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTT
YPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTY
-bash: PuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPu
TTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYP
uTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTY
PuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTT
YPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuTTYPuT
TYPuTTYPuTTYPuTTYPuTTY: command not found
/home/cis90/simben $
```

*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 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,  US
*/



static char rcsid[]="$Id: gcal.(



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

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

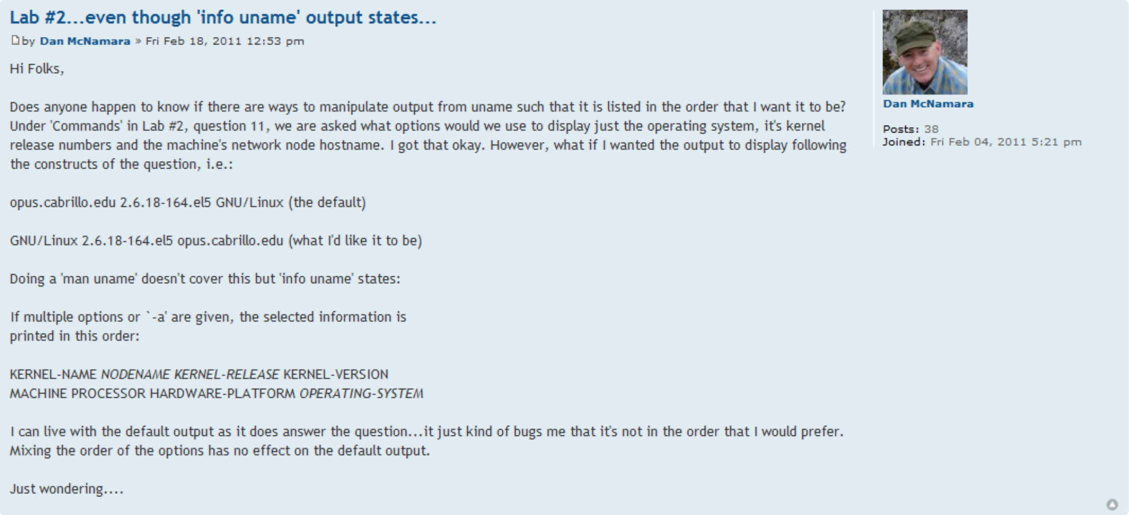
```
[rsimms@nosmo src]$ file /usr/bin/cal
/usr/bin/cal: ELF 32-bit LSB executable, Intel 80386, version 1
(SYSV), for GNU/Linux 2.2.5, dynamically linked (uses shared lib
s), stripped
[rsimms@nosmo src]$ 
```

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, it's 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?
   Type your answer in the chat window.

2) Is the **scavenge** command a binary executable or a shell script?
   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 bc as the argument.
   Type your answer in the chat window.

Inputs to commands

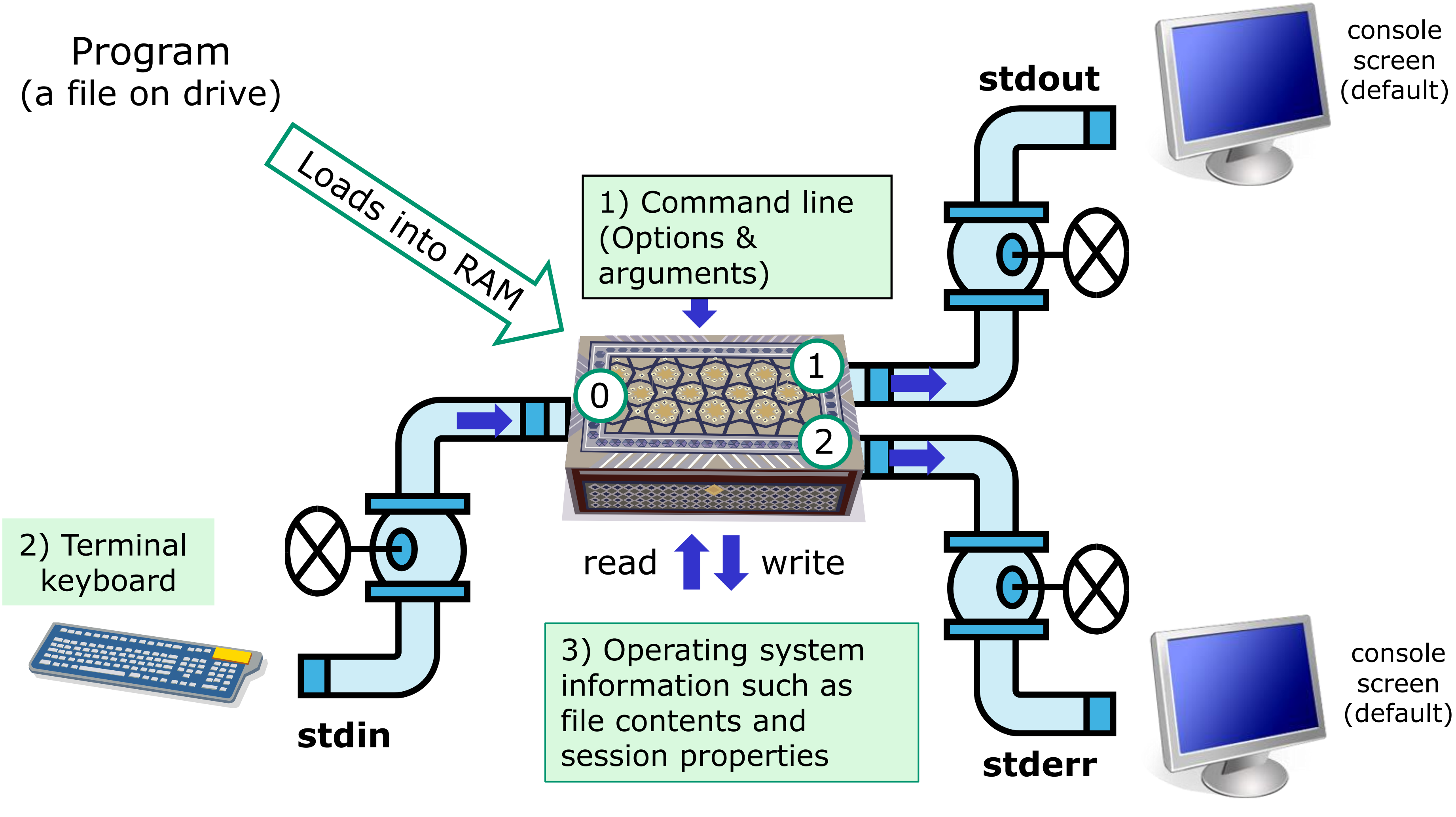
You will get these questions when you submit Lab 2

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



Program
(a file on drive)

Loads into RAM

1) Command line
(Options &
arguments)

2) Terminal
keyboard

3) Operating system
information such as
file contents and
session properties

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

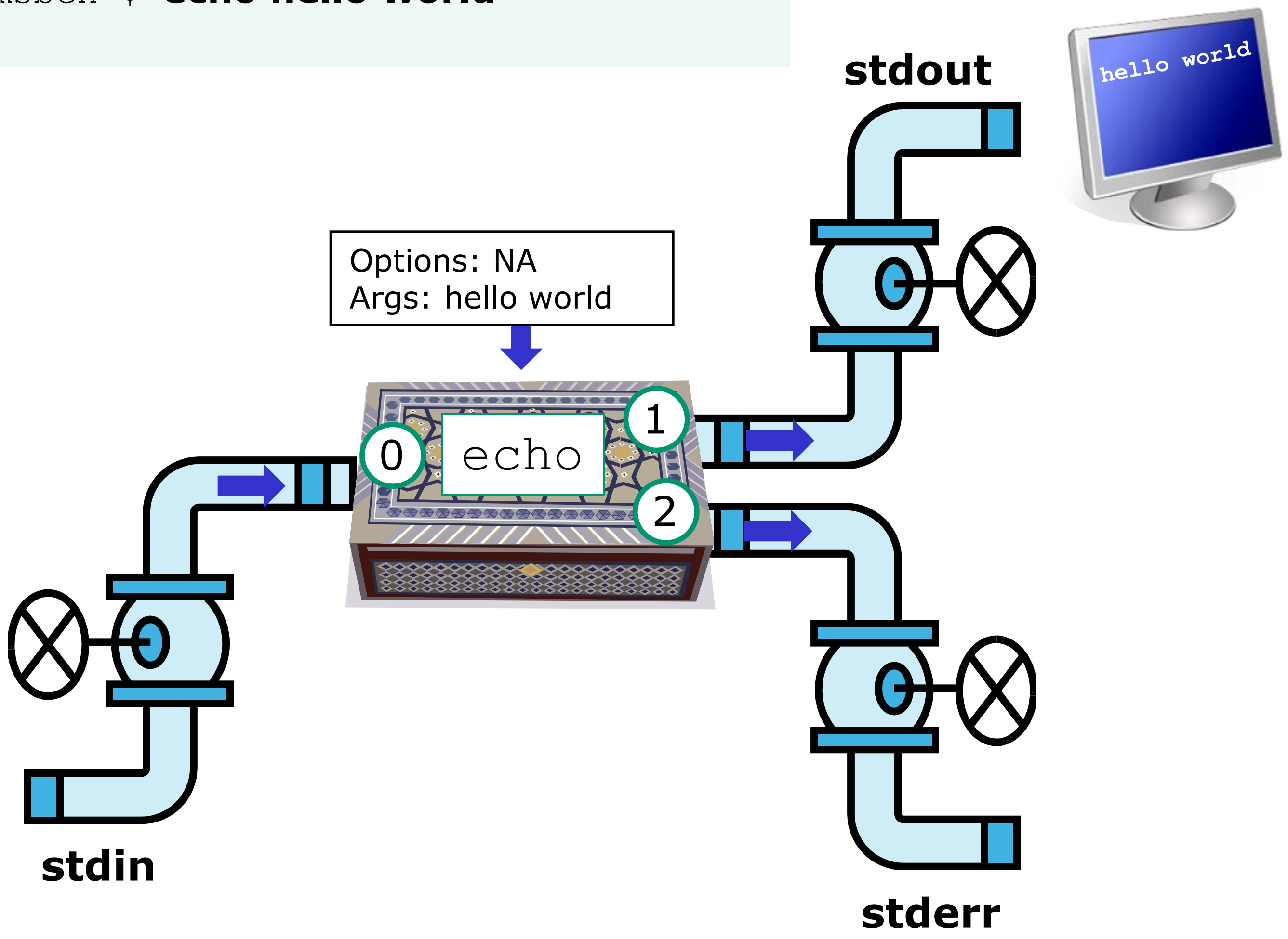
```
/home/cis90/simmen $ echo hello world
hello world
```

```
/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
```



Options: NA
Args: 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?

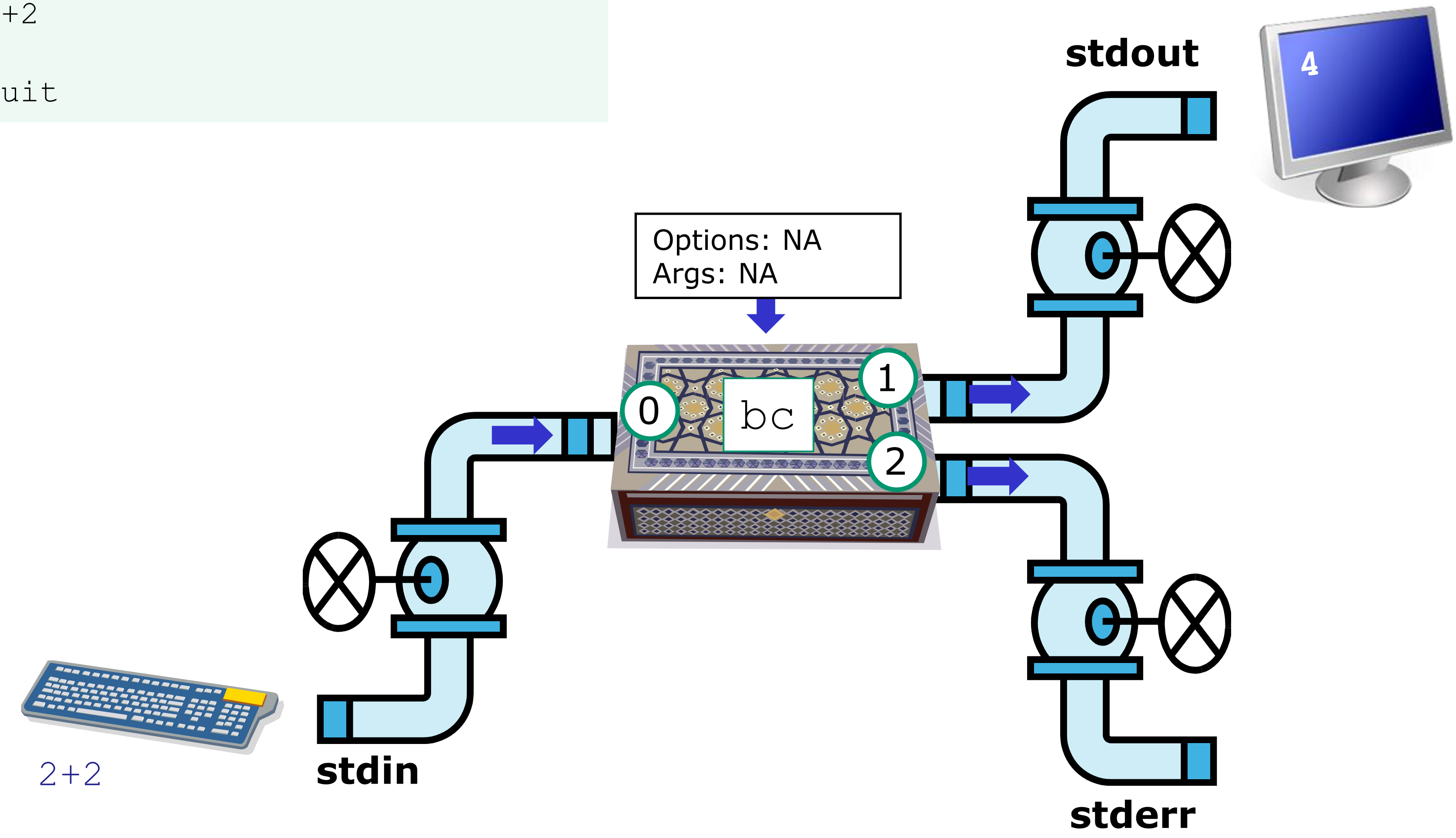
```
/home/cis90/simmsben $ bc
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free
Software Foundation, Inc.
This is free software with ABSOLUTELY NO
WARRANTY.
For details type `warranty'.
2+2
4
500-200+3
303
sqrt(64)
8
quit
```

```
/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?

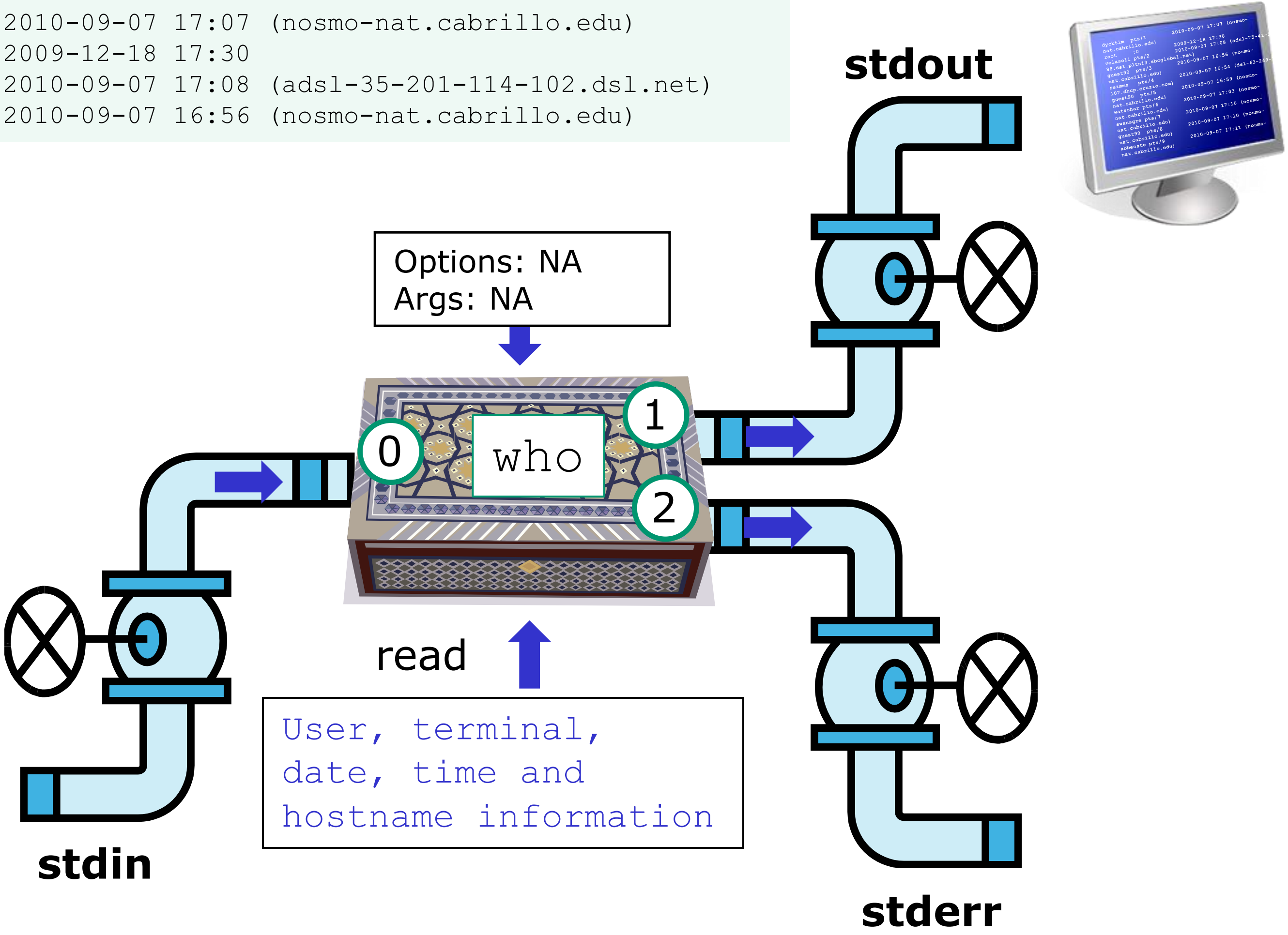
```
/home/cis90/simmen $ who
dycktim  pts/1        2010-09-07 17:07 (nosmo-nat.cabrillo.edu)
root     :0           2009-12-18 17:30
velasoli pts/2        2010-09-07 17:08 (adsl-35-201-114-102.dsl.net)
guest90  pts/3        2010-09-07 16:56 (nosmo-nat.cabrillo.edu)
rsimms   pts/4        2010-09-07 15:54 (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)
```



User, terminal, date, time and hostname information

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

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`

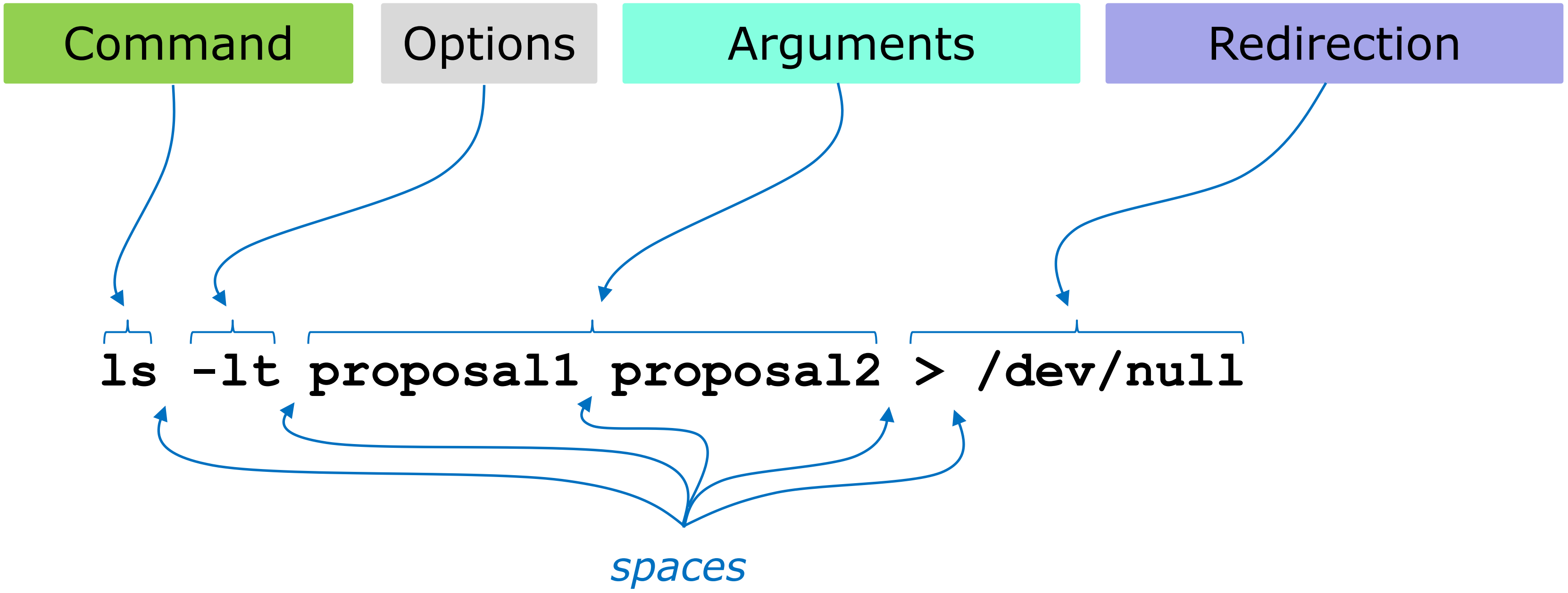
Arguments – follow the options. Multiple arguments must be separated by spaces.

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



Command

Options

Arguments

Redirection

```
ls -lt proposal1 proposal2 > /dev/null
```

spaces

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 proposal1 timecal
proposal1: 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

Command Syntax

Command	Options	Arguments	Redirection

```
/home/cis90/simben $ ls -l -i -a /bin Poems/ letter small_town > /dev/null
/home/cis90/simben $
```

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 -l -i -a /bin Poems/ letter small_town > /dev/null
/home/cis90/simben $
```

Please parse the command line above

Command: ls

Options:

How many: 3
What are they: l, i, a

Arguments:

How many: 4
What are they: /bin, Poems/, letter, small_town

Redirection:

How many: 1
What is redirected: stdout redirected to /dev/null

Command Syntax

Command | Options | Arguments | Redirection

```
/home/cis90/simben $ echo "1   2   3   4   5"
1   2   3   4   5
```

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 "1  2  3  4  5"
1  2  3  4  5
```

Please parse the command line above

Command: echo

Options:
How many: NA
What are they: NA

Arguments:
How many: 1
What are they: "1 2 3 4 5"

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*

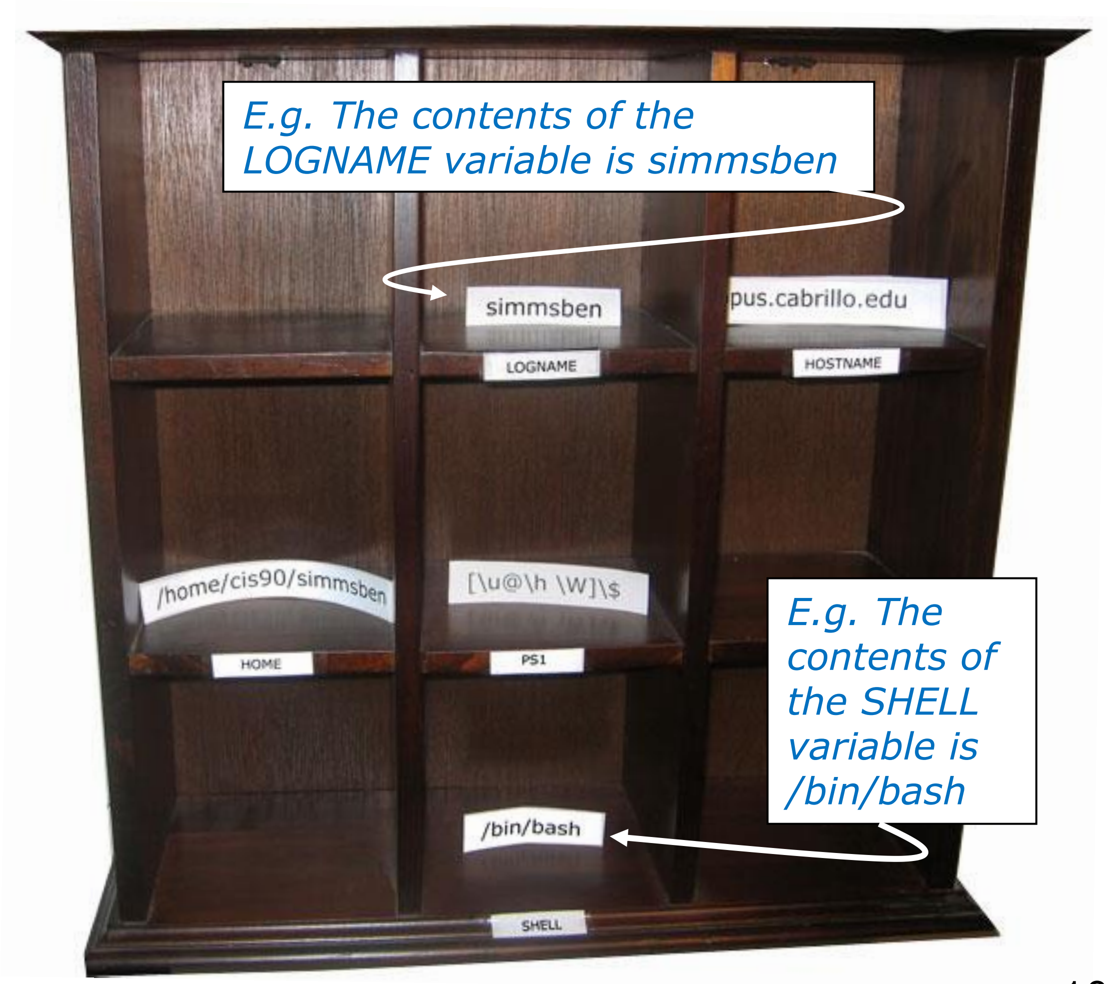
To change the value of a variable, use an = sign with no surrounding blanks and no $

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

Variables

Variables are stored in memory. You can think of variables as named boxes containing data.

```
$ echo $LOGNAME
simmsben

$ echo $HOSTNAME
opus.cabrillo.edu

$ echo $HOME
/home/cis90/simmsben

$ echo $SHELL
/bin/bash
```



E.g. The contents of the LOGNAME variable is simmsben

E.g. The contents of the SHELL variable is /bin/bash

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 cd command (with no arguments)
LOGNAME	User's username for logging in with.
PATH	List of directories, separated by :'s, for the Shell to search for commands (which are program files) .
PS1	The prompt string.
PWD	Current working directory
SHELL	Name of the Shell program being used.
TERM	Type of terminal device , e.g. dumb, vt100, xterm, ansi, 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
```

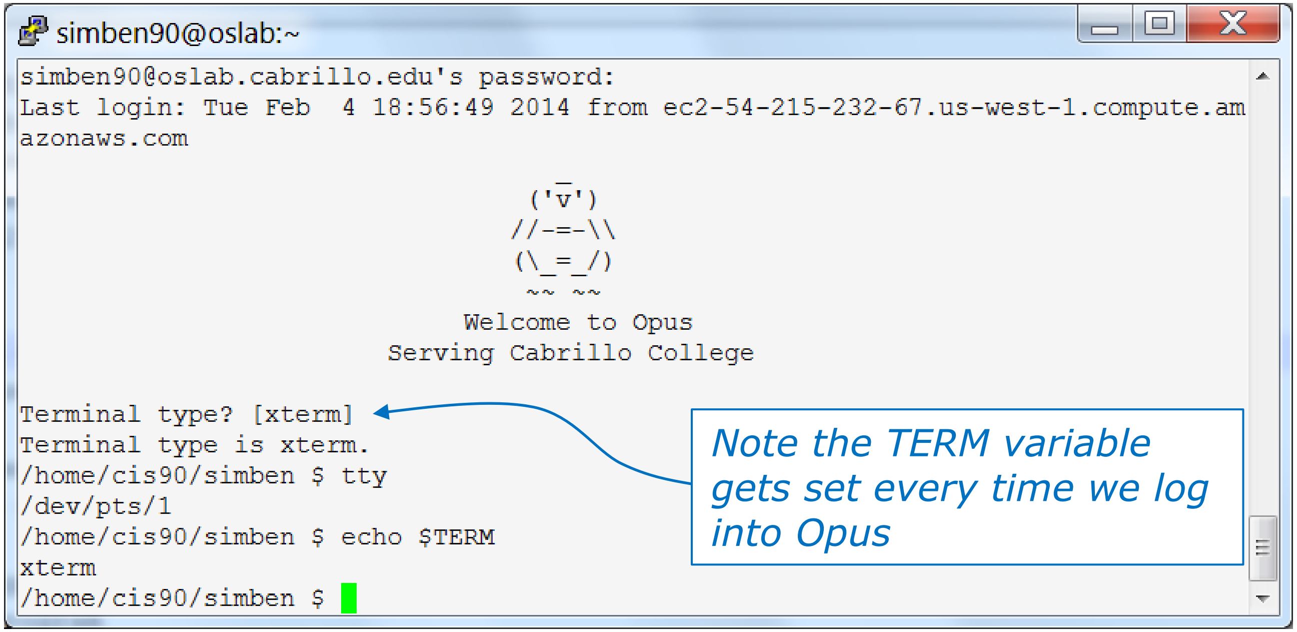
Shows your shell

```
/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

The TERM variable

The TERM variable holds the terminal __type__ which is different than the terminal __device__



```
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 $
```

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

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

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

The SHELL variable

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

The SHELL variable will be set to the name of the shell your 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

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 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

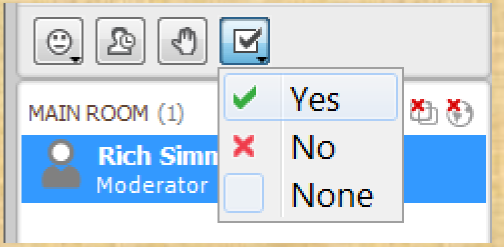
Class Exercise

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 x if stuck on CCC Confer



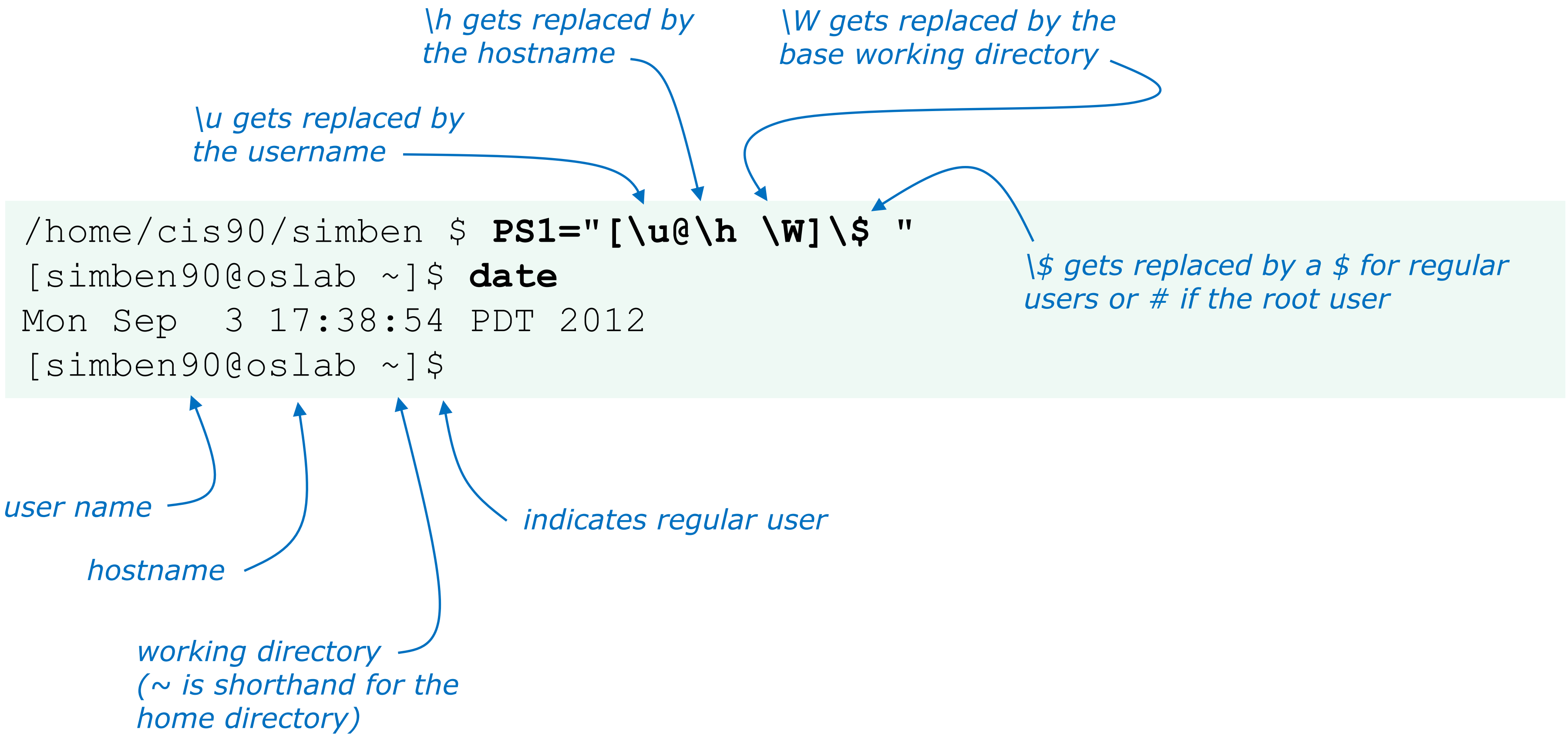
Need a fresh start -- just log out and back in again and your prompt will be back to normal!

Changing the shell prompt

(PS1 variable)

Changing the prompt

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



```
/home/cis90/simben $ PS1="[\u@\h \W]\$ "
[simben90@oslab ~]$ date
Mon Sep  3 17:38:54 PDT 2012
[simben90@oslab ~]$
```

\u gets replaced by the username

\h gets replaced by the hostname

\W gets replaced by the base working directory

\$ 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@\$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_fignore:hostco
mplete: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
3;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:*.lzma=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

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

```
/home/cis90/simben $ env
HOSTNAME=oslab.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:*.lzma=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:*.mp4
v=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=0
1;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 $
```

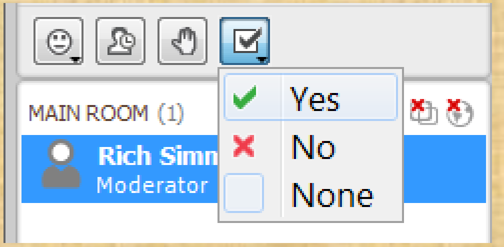
Class Exercise

PS1 "Prompt" variable

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

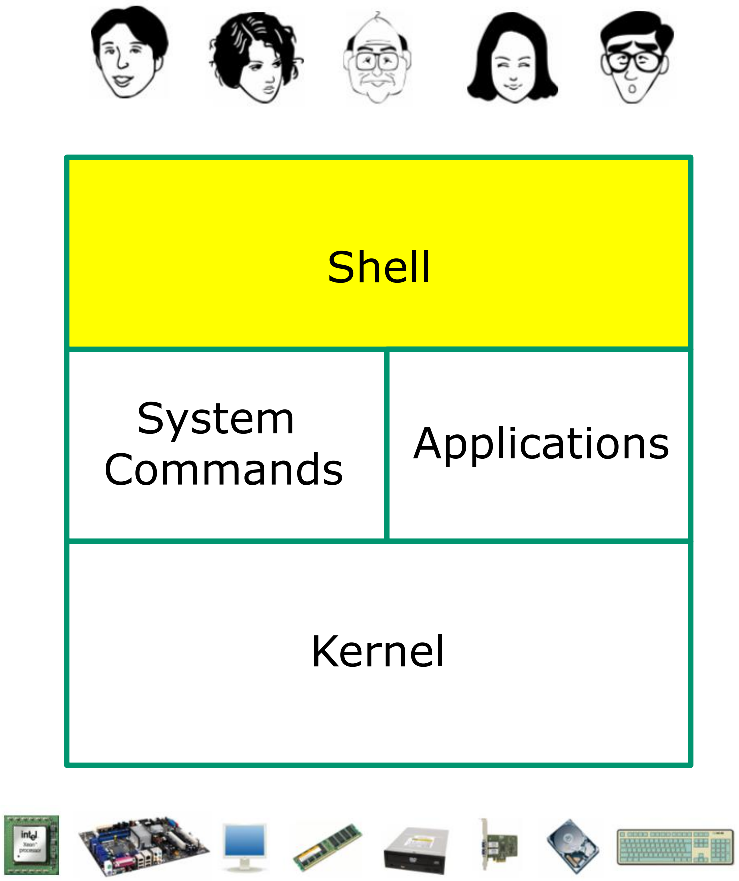
Include a space after the ?

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



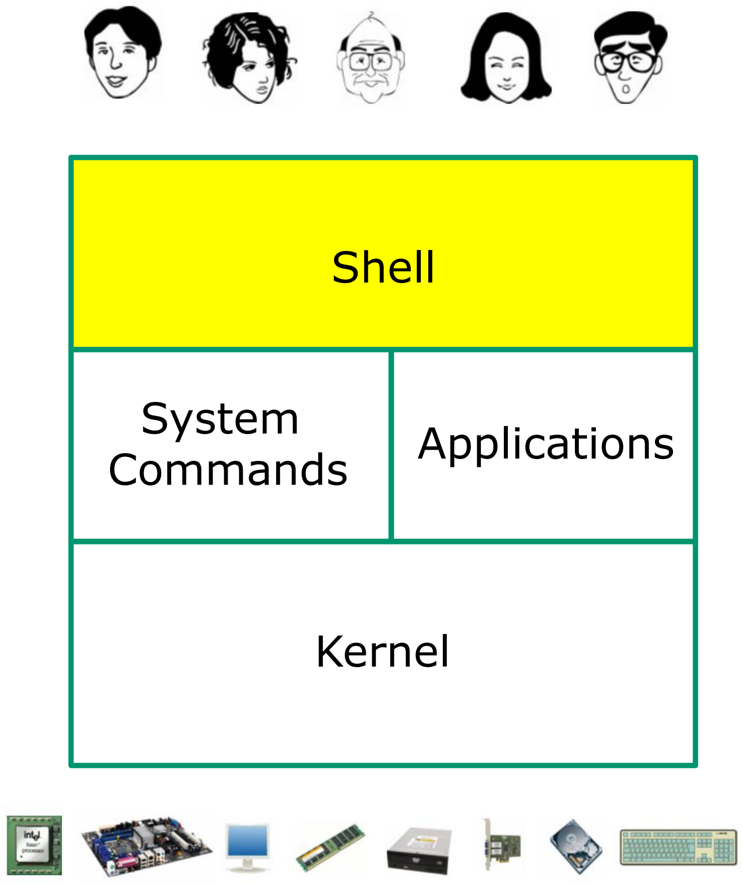
The Shell (Deep Dive)

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** (born 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



Shell

System Commands

Applications

Kernel

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

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

2) Parse command user typed

Example:

```
ls -lt proposal1 proposal2
```

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

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

Life of the Shell

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

3) Search path for the program to run

```
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
     .
```

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
```

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

Life of the Shell

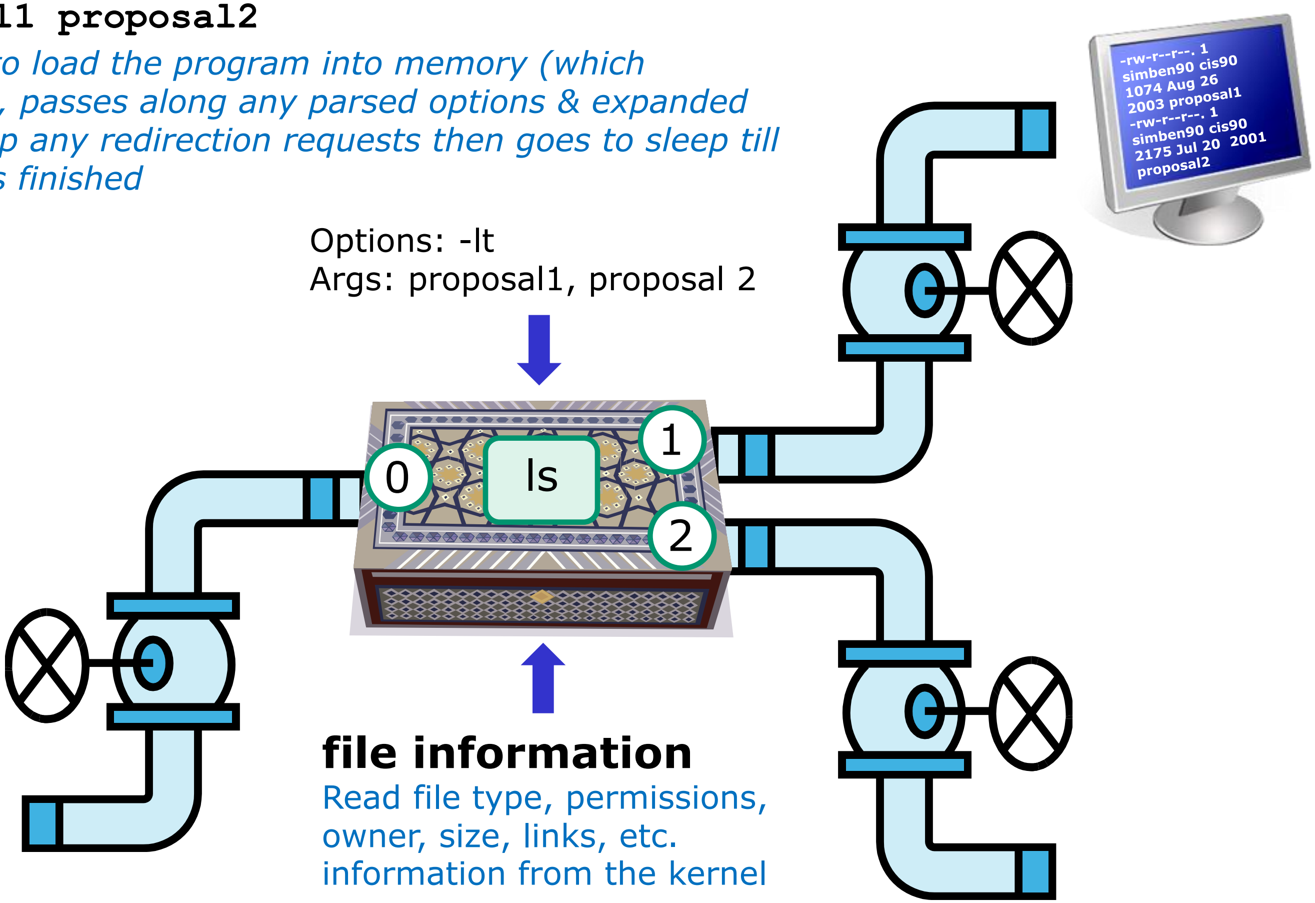
Shell Steps

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



```
-rw-r--r--. 1
simben90 cis90
1074 Aug 26
2003 proposal1
-rw-r--r--. 1
simben90 cis90
2175 Jul 20  2001
proposal2
```

Options: -lt
Args: proposal1, proposal 2

file information

Read file type, permissions, owner, size, links, etc. information from the kernel

Life of the Shell

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

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)

```
/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?

A) Capital L typo, bash; B) non-existent proposal 5, ls; C) incomplete w option, ls;
D) no space between arguments, ls; E) no space after command, bash

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 <u>allow</u> variable expansion
- Single ' quotes <u>block</u> 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 <u>weak</u> quotes because they allow the shell to expand variables. Single quotes are called <u>strong</u> quotes because they block the shell from expanding variables.

Metacharacters - quotes

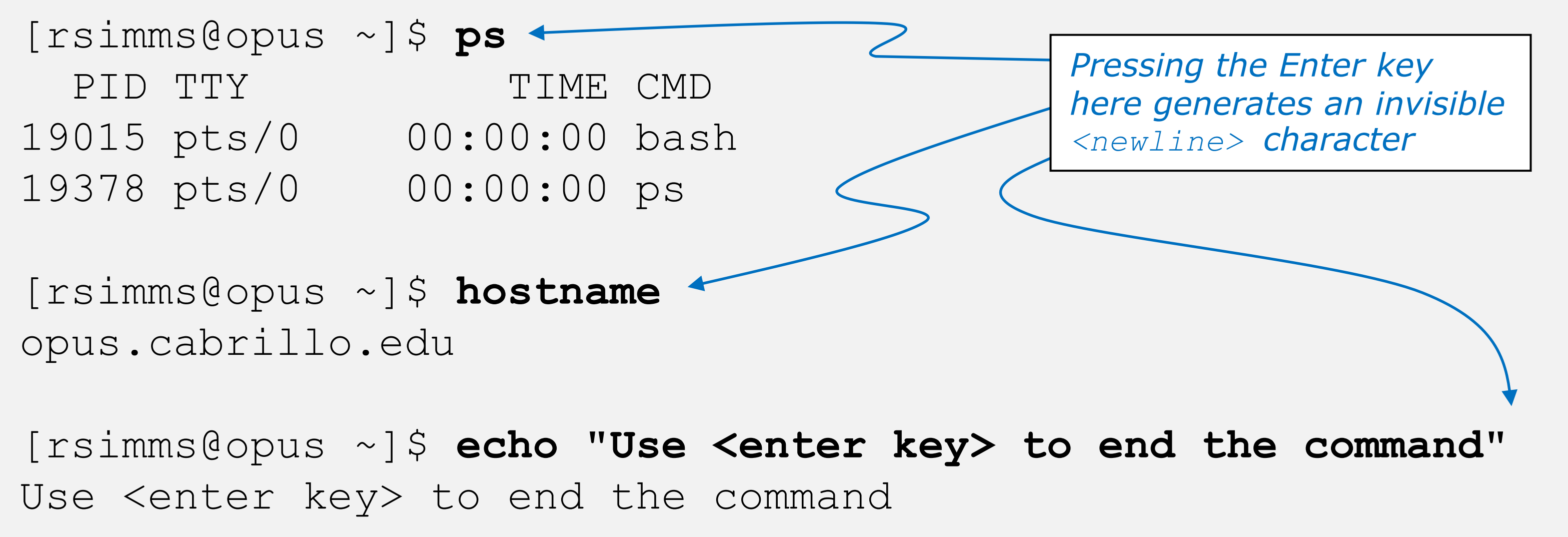
```
/home/cis90/simben $ echo '"double quotes"'
"double quotes"
```

```
/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
```

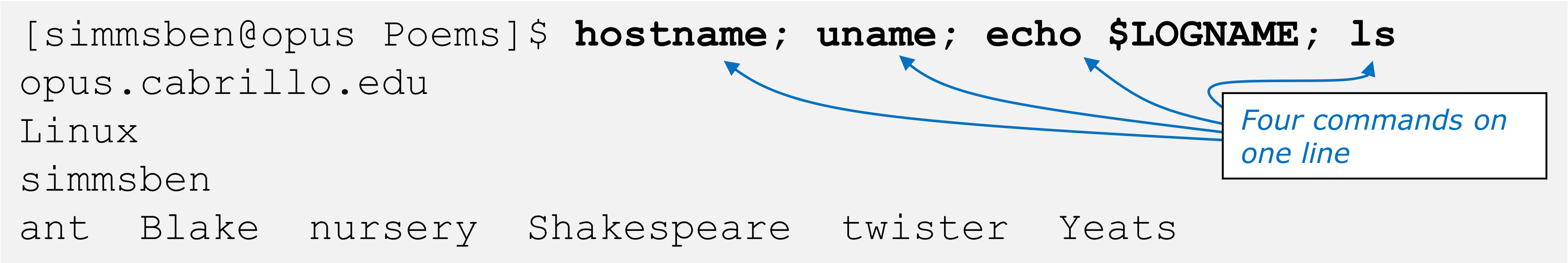
Escape the $ (which shows the value of the variable)

```
[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
```

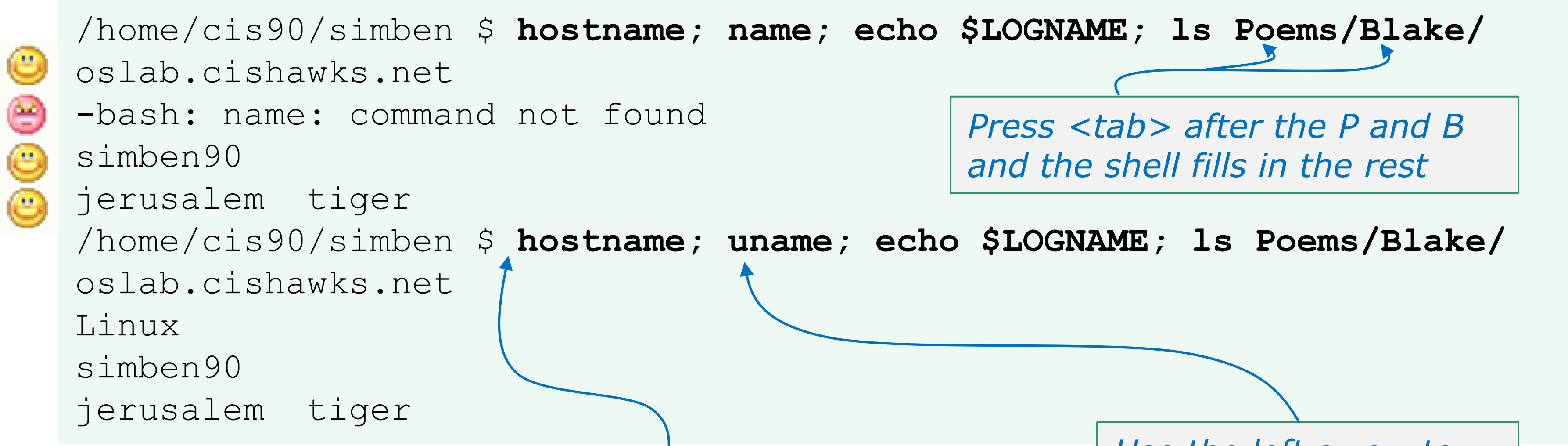
Four commands on one line

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/c
is90/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

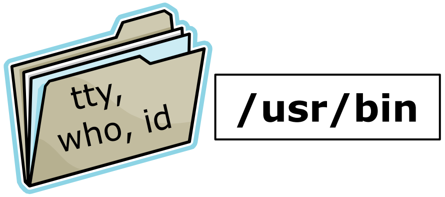
Experiment – Breaking the Path

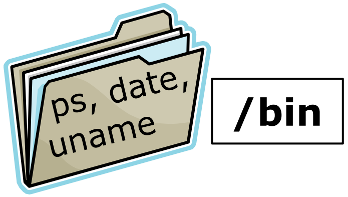
*The **echo** command is built into bash*

```
/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*





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 $
```

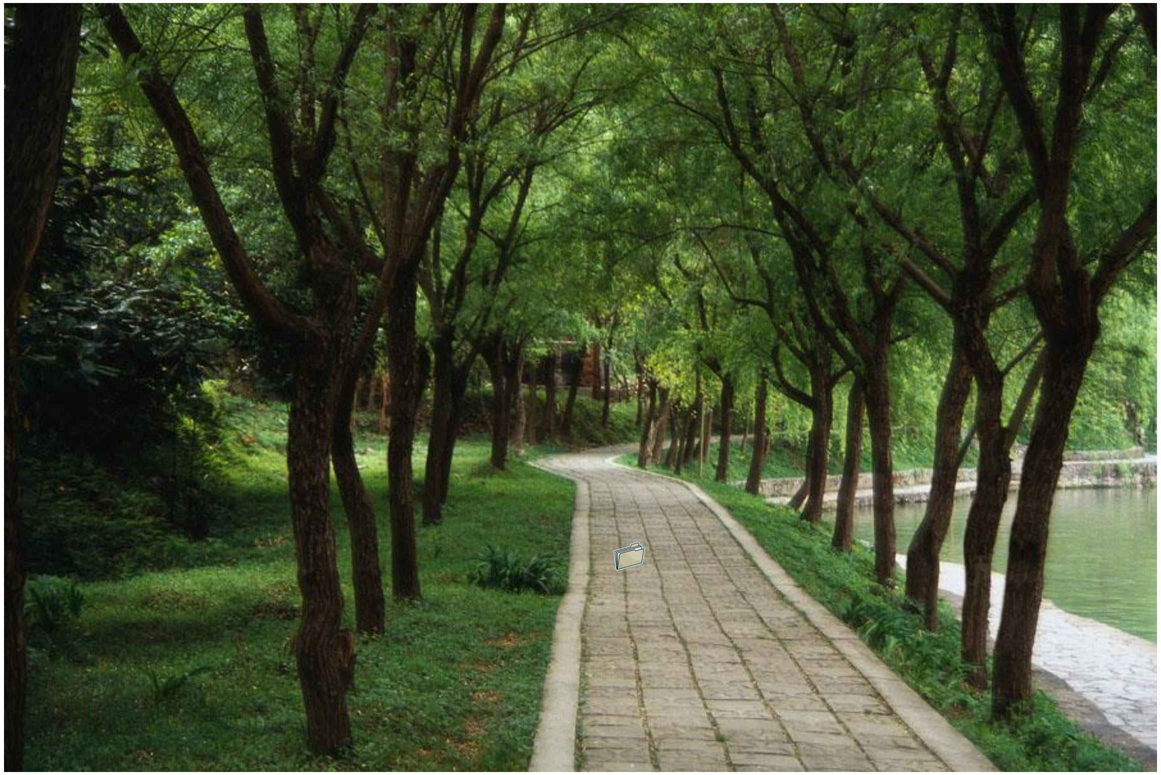
Break the path by setting it to null

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 $

/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
```

Add the /bin directory to the path

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

***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 $

/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
```

Append the /usr/bin directory to the path

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 $
```

tty,
who, id
/usr/bin
ps, date,
uname
/bin

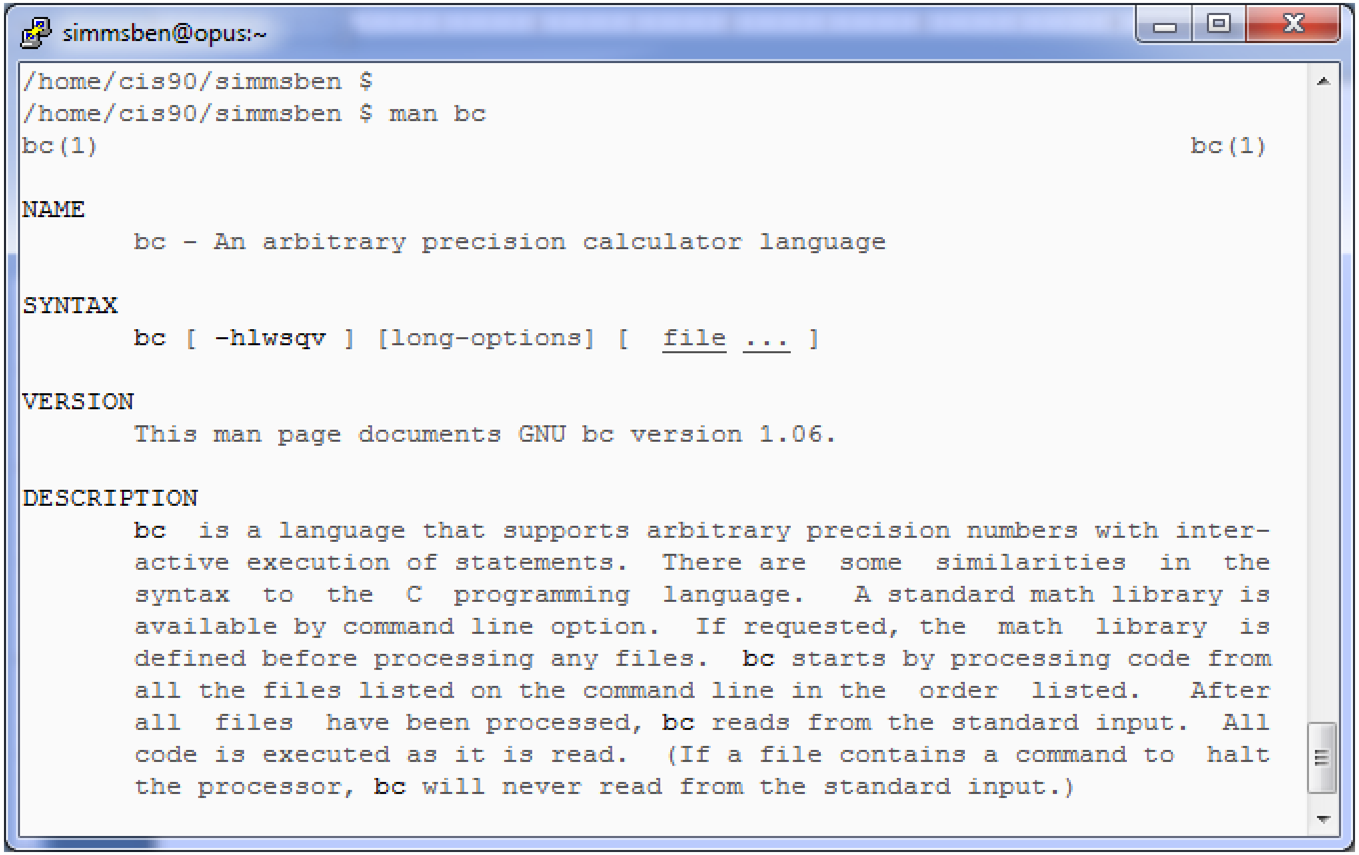
Need a fresh start -- just log out and back in again and your path will be back to normal!

Docs

Using man (manual) pages

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

Example: **man bc**



```
/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.)
```

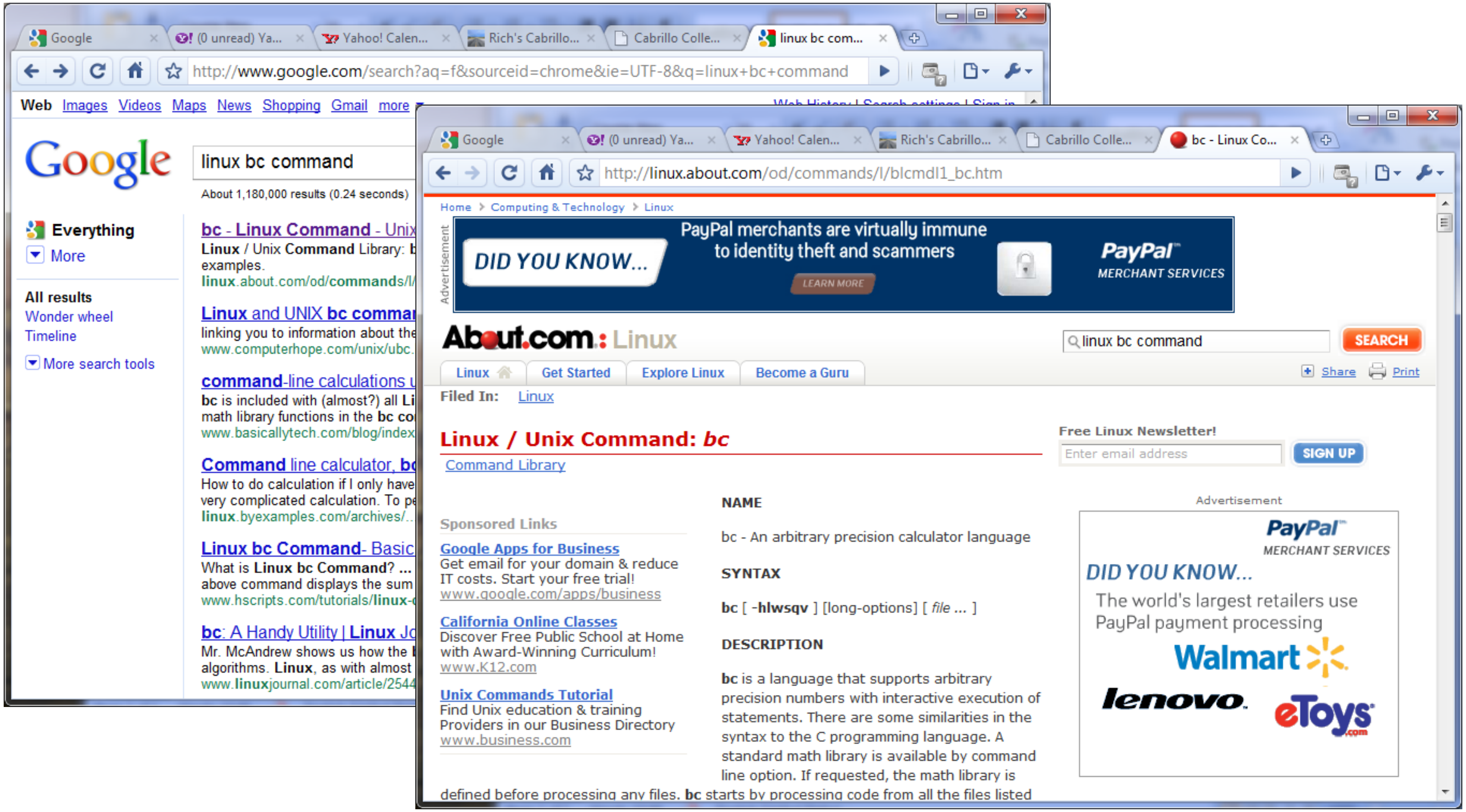
Use these keys to scroll

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



http://www.google.com/search?aq=f&sourceid=chrome&ie=UTF-8&q=linux+bc+command

Web Images Videos Maps News Shopping Gmail more
Web History | Search settings | Sign in
Google

linux bc command

About 1,180,000 results (0.24 seconds)

bc - Linux Command - Unix
Linux / Unix **Command** Library: b
examples.
linux.about.com/od/commands/l/

Linux and UNIX bc comma
linking you to information about the
www.computerhope.com/unix/ubc.

command-line calculations u
bc is included with (almost?) all **Li**
math library functions in the **bc co**
www.basicallytech.com/blog/index

Command line calculator, bc
How to do calculation if I only have
very complicated calculation. To pe
linux.byexamples.com/archives/...

Linux bc Command- Basic
What is **Linux bc Command**? ...
above command displays the sum
www.hscripts.com/tutorials/linux-c

bc: A Handy Utility | Linux Jo
Mr. McAndrew shows us how the
algorithms. **Linux**, as with almost
www.linuxjournal.com/article/2544

http://linux.about.com/od/commands/l/blcmdl1_bc.htm

Filed In: Linux

Linux / Unix Command: bc

Command Library

Free Linux Newsletter!

Enter email address SIGN UP

NAME

bc - An arbitrary precision calculator language

SYNTAX

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

DESCRIPTION

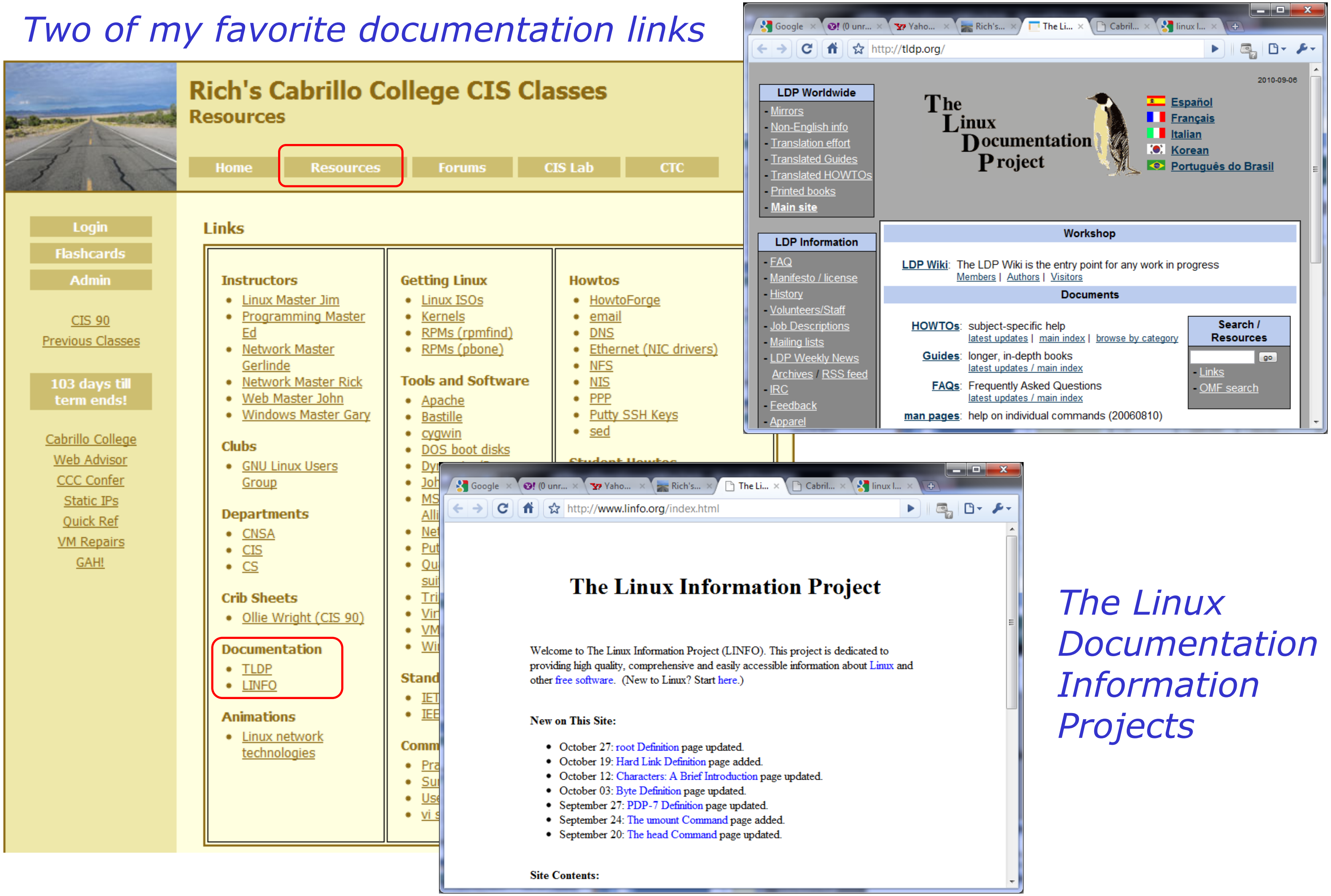
bc is a language that supports arbitrary precision numbers with interactive execution of statements. There are some similarities in the syntax to the C programming language. A standard math library is available by command line option. If requested, the math library is defined before processing any files.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

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
- Dy
- Joh
- MS
- Alli
- Net
- Put
- Qu sui
- Tri
- Vir
- VM
- Wi

Stand
- IET
- IEE

Comm
- Pra
- Su
- Use
- vi s

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

Student Howtos

Google | (0 unr... | Yaho... | Rich's... | The Li... | Cabril... | linux l...

http://tldp.org/

2010-09-06

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

The Linux Documentation Project

Español
Français
Italian
Korean
Português do Brasil

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

go
- Links
- OMF search

Google | (0 unr... | Yaho... | Rich's... | The Li... | Cabril... | linux l...

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:

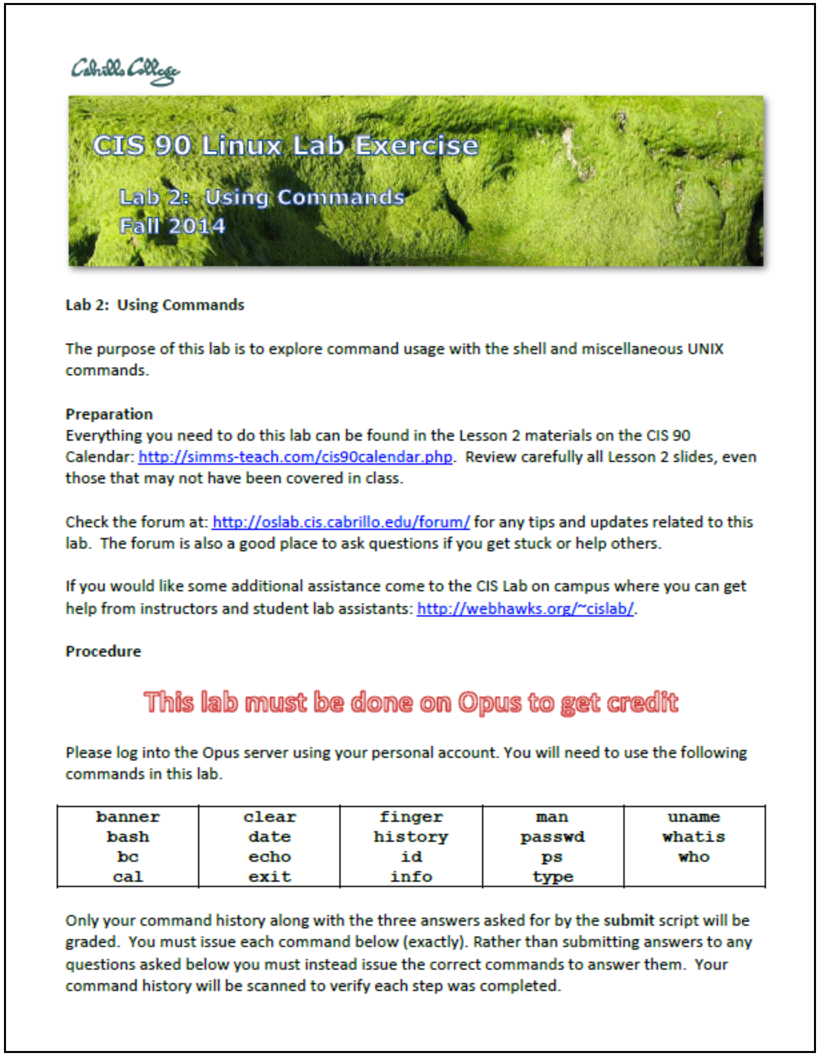
- October 27: root Definition page updated.
- October 19: Hard Link Definition page added.
- October 12: Characters: A Brief Introduction page updated.
- October 03: Byte Definition page updated.
- September 27: PDP-7 Definition page updated.
- September 24: The umount Command page added.
- September 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 bash bc cal	clear date echo exit	finger history id info	man passwd ps type	uname whatis who

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.

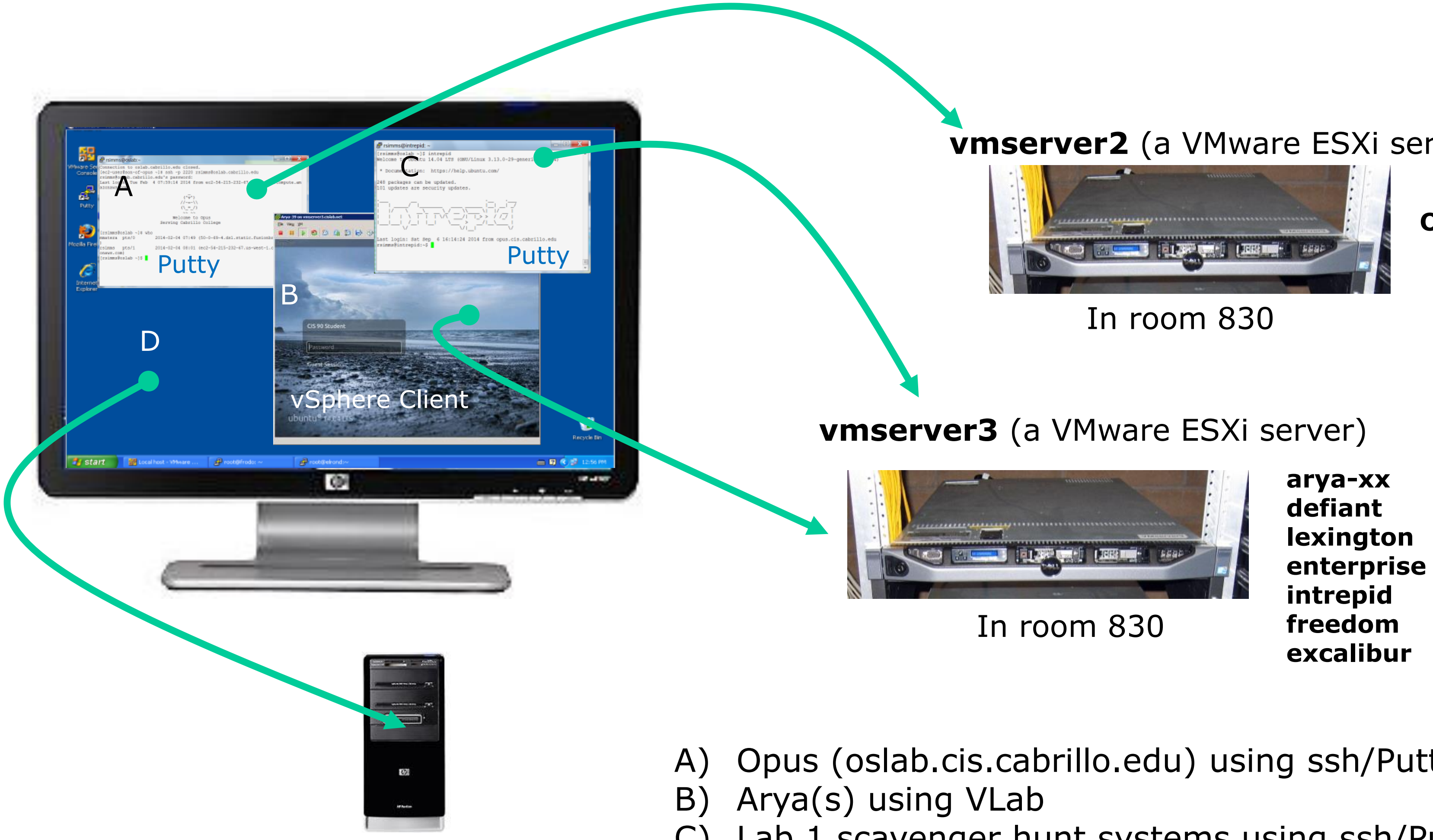
Lab #2

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



vmserver2 (a VMware ESXi server)

Opus

In room 830

vmserver3 (a VMware ESXi server)

arya-xx
defiant
lexington
enterprise
intrepid
freedom
excalibur

In room 830

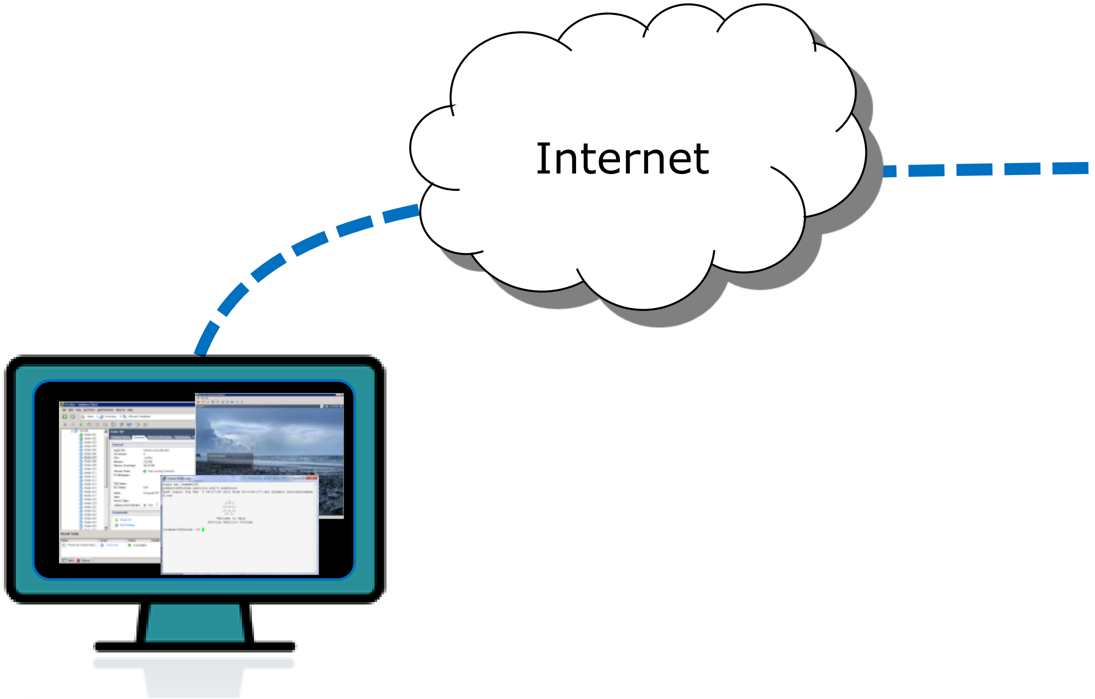
Your Computer
In the lab or at home

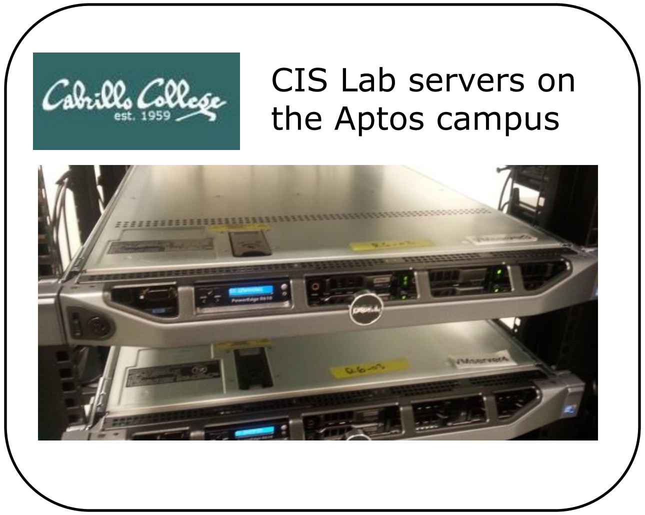
A) Opus (oslab.cis.cabrillo.edu) using ssh/Putty
B) Arya(s) using VLab
C) Lab 1 scavenger hunt systems using ssh/Putty
D) Your computer's desktop

Using CIS VLab (Virtual Lab)

Third driving lesson

Accessing CIS VLab VMs



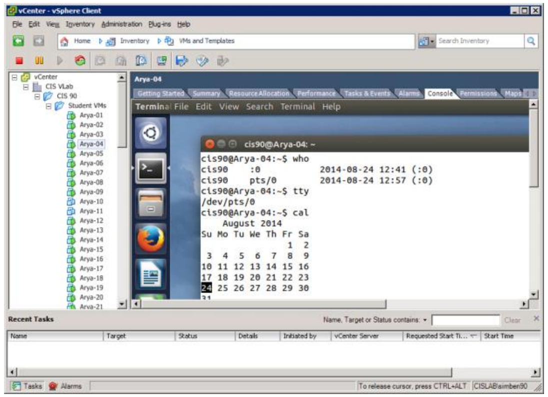


CIS Lab servers on the Aptos campus

Home

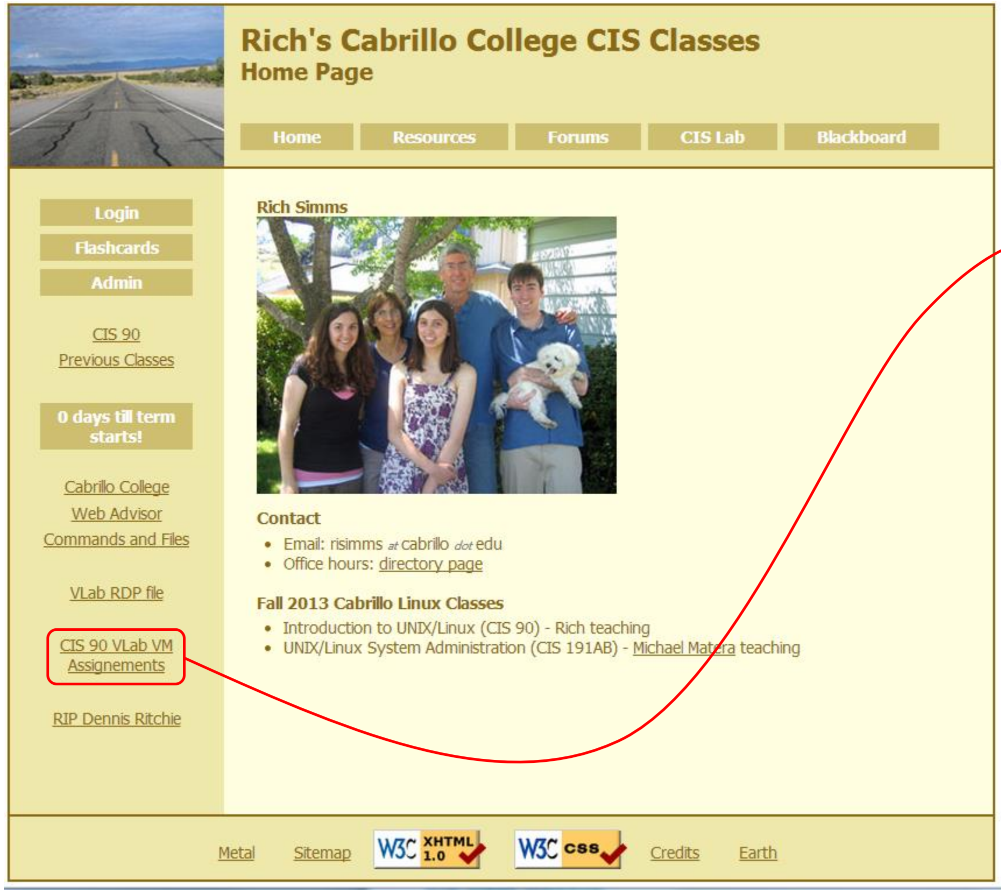
School

Travel


vCenter - vSphere Client
File Edit View Inventory Administration Plug-ins Help
Home Inventory VMs and Templates
vCenter
CIS VLab
CIS 90
Student VMs
Arya-01 … Arya-21
Arya-04
Getting Started Summary Resource Allocation Performance Tasks & Events Alarms Console Permissions Maps
Terminal File Edit View Search Terminal Help
cis90@Arya-04: ~
Recent Tasks
Name, Target or Status contains: Clear
Name Target Status Details Initiated by vCenter Server Requested Start Ti... Start Time
Tasks Alarms
To release cursor, press CTRL+ALT

```
cis90@Arya-04:~$ who
cis90    :0           2014-08-24 12:41 (:0)
cis90    pts/0        2014-08-24 12:57 (:0)
cis90@Arya-04:~$ tty
/dev/pts/0
cis90@Arya-04:~$ cal
    August 2014
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
31
```

vmware



Rich's Cabrillo College CIS Classes
Home Page

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

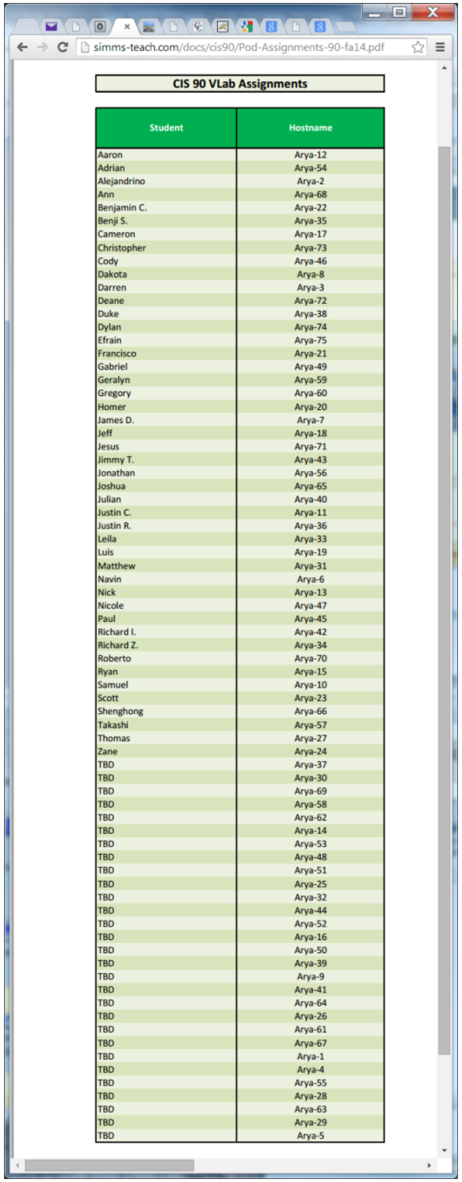
Contact

- Email: risimms *at* cabrillo *dot* 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 Matera teaching

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

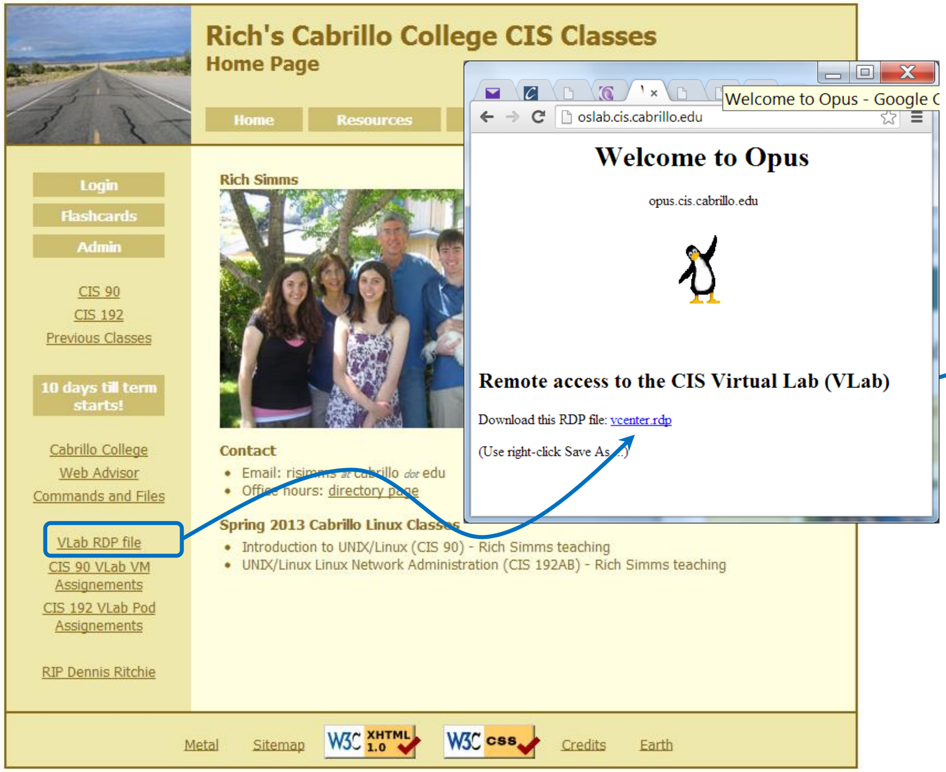


simms-teach.com/docs/cis90/Pod-Assignments-90-fa14.pdf

CIS 90 VLab Assignments

Student	Hostname
Aaron	Arya-12
Adrian	Arya-54
Alejandrino	Arya-2
Ann	Arya-68
Benjamin C.	Arya-22
Benji S.	Arya-35
Cameron	Arya-17
Christopher	Arya-73
Cody	Arya-46
Dakota	Arya-8
Darren	Arya-3
Deane	Arya-72
Duke	Arya-38
Dylan	Arya-74
Efrain	Arya-75
Francisco	Arya-21
Gabriel	Arya-49
Geralyn	Arya-59
Gregory	Arya-60
Homer	Arya-20
James D.	Arya-7
Jeff	Arya-18
Jesus	Arya-71
Jimmy T.	Arya-43
Jonathan	Arya-56
Joshua	Arya-65
Julian	Arya-40
Justin C.	Arya-11
Justin R.	Arya-36
Leila	Arya-33
Luis	Arya-19
Matthew	Arya-31
Navin	Arya-6
Nick	Arya-13
Nicole	Arya-47
Paul	Arya-45
Richard I.	Arya-42
Richard Z.	Arya-34
Roberto	Arya-70
Ryan	Arya-15
Samuel	Arya-10
Scott	Arya-23
Shenghong	Arya-66
Takashi	Arya-57
Thomas	Arya-27
Zane	Arya-24
TBD	Arya-37
TBD	Arya-30
TBD	Arya-69
TBD	Arya-58
TBD	Arya-62
TBD	Arya-14
TBD	Arya-53
TBD	Arya-48
TBD	Arya-51
TBD	Arya-25
TBD	Arya-32
TBD	Arya-44
TBD	Arya-52
TBD	Arya-16
TBD	Arya-50
TBD	Arya-39
TBD	Arya-9
TBD	Arya-41
TBD	Arya-64
TBD	Arya-26
TBD	Arya-61
TBD	Arya-67
TBD	Arya-1
TBD	Arya-4
TBD	Arya-55
TBD	Arya-28
TBD	Arya-63
TBD	Arya-29
TBD	Arya-5

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

RIP Dennis Ritchie

Rich Simms

Contact
- Email: risimms at cabrillo dot edu
- Office hours: directory page

Spring 2013 Cabrillo Linux Classes
- Introduction to UNIX/Linux (CIS 90) - Rich Simms teaching
- UNIX/Linux Linux Network Administration (CIS 192AB) - Rich Simms teaching

Metal Sitemap W3C XHTML 1.0 W3C CSS Credits Earth

Welcome to Opus - Google C
oslab.cis.cabrillo.edu

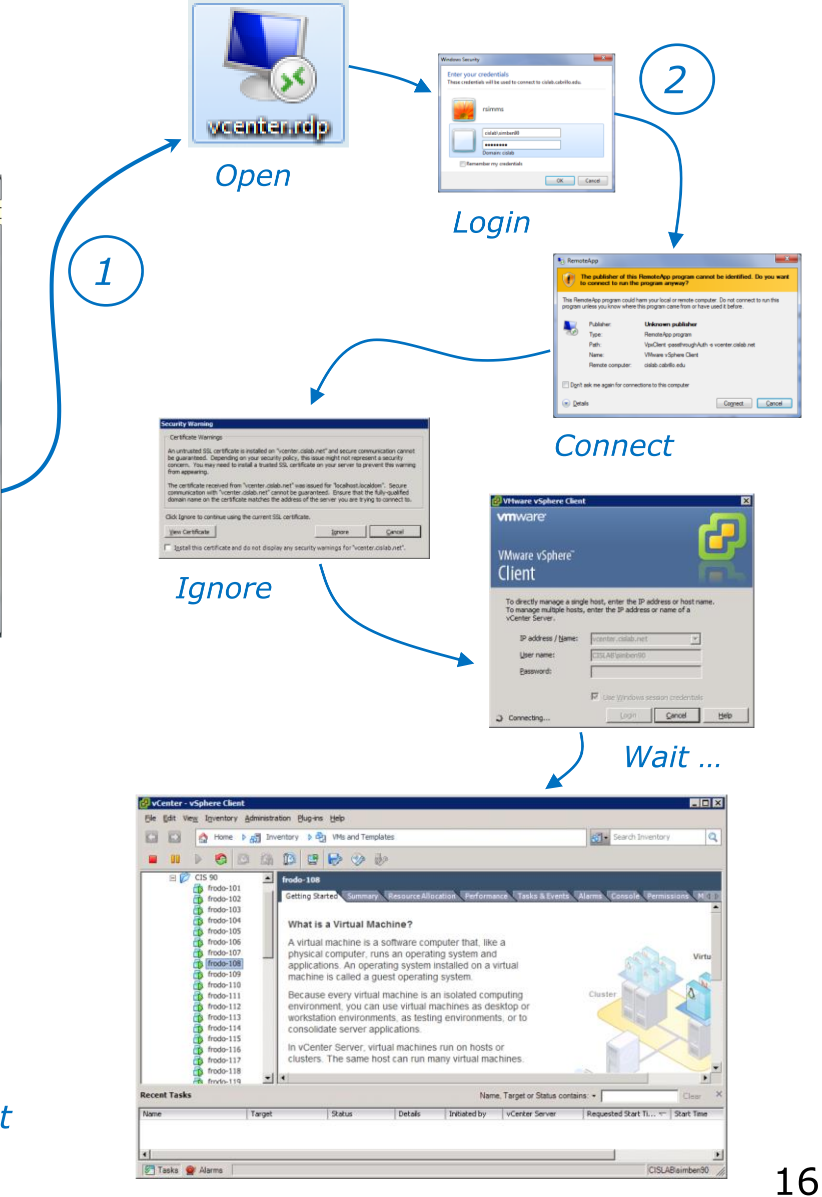
Welcome to Opus

opus.cis.cabrillo.edu

Remote access to the CIS Virtual Lab (VLab)

Download this RDP file: vcenter.rdp

(Use right-click Save As ...)



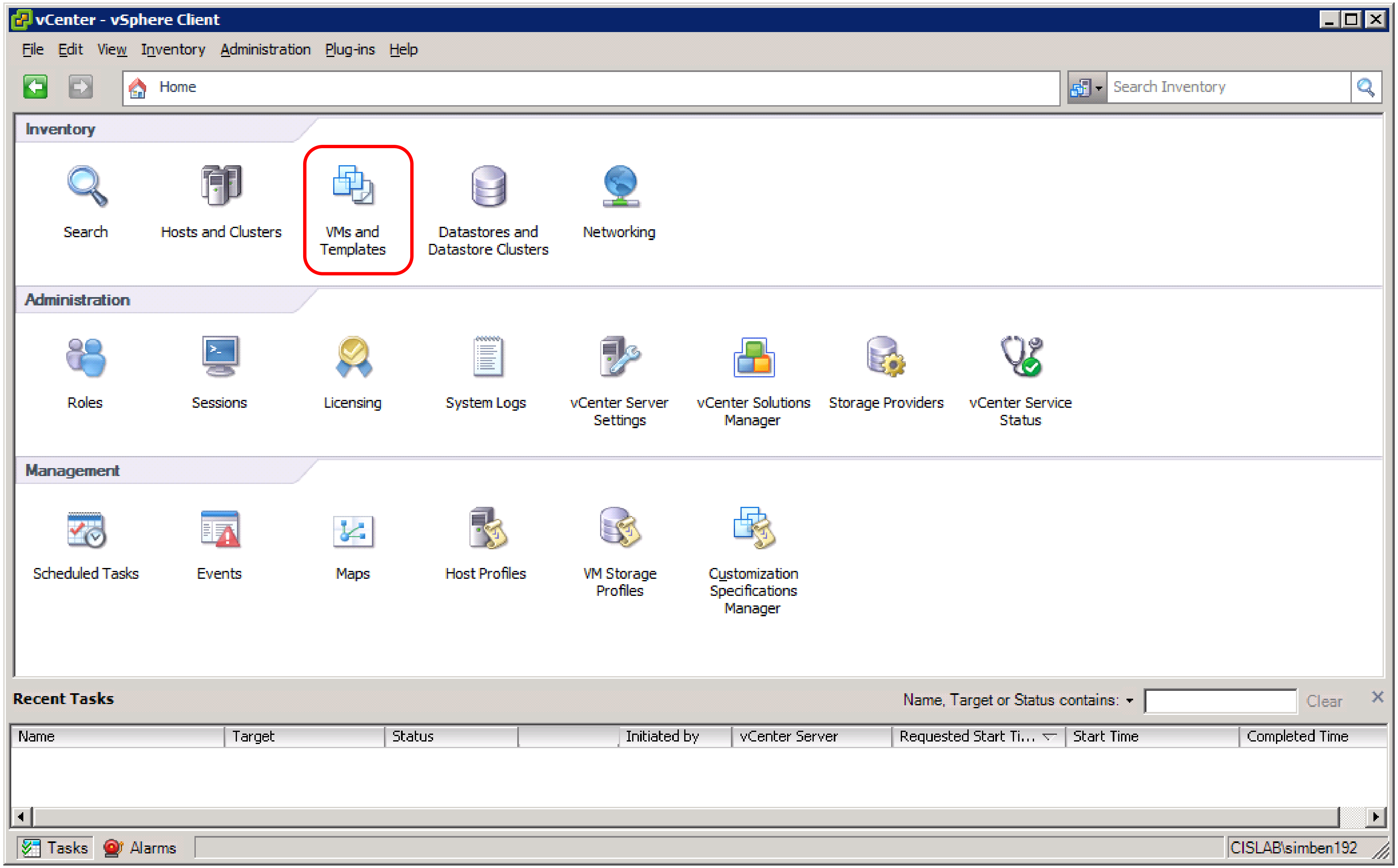
Locate and select your assigned VM

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

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

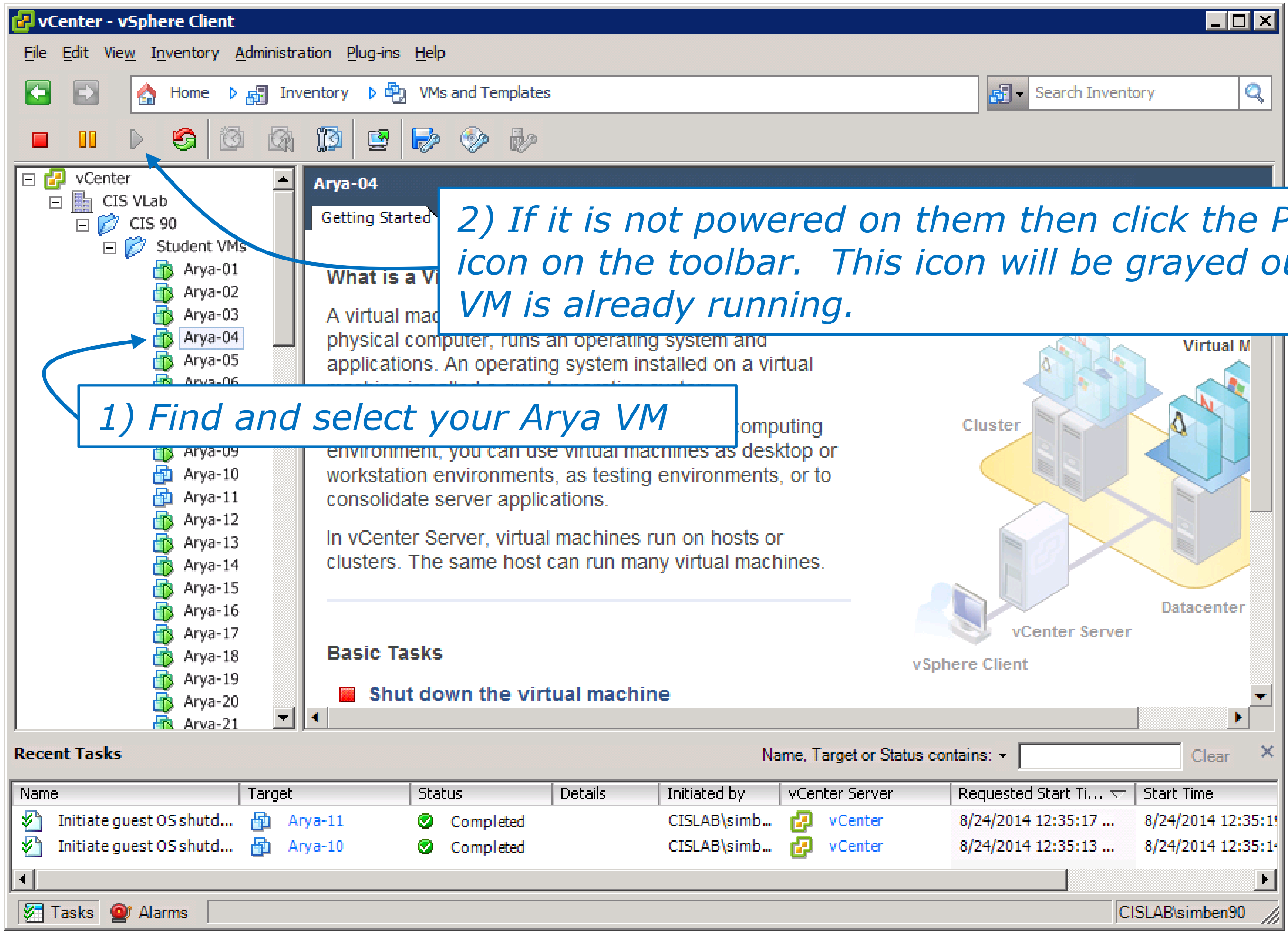
3) When entering your username and password you must preface your username with the "cislab\", for example Benji would use: cislab\simben90

CIS VLab Home View



Click VMs and Templates to get to your course VMs

Selecting and powering on your VM



What is a Vi

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

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

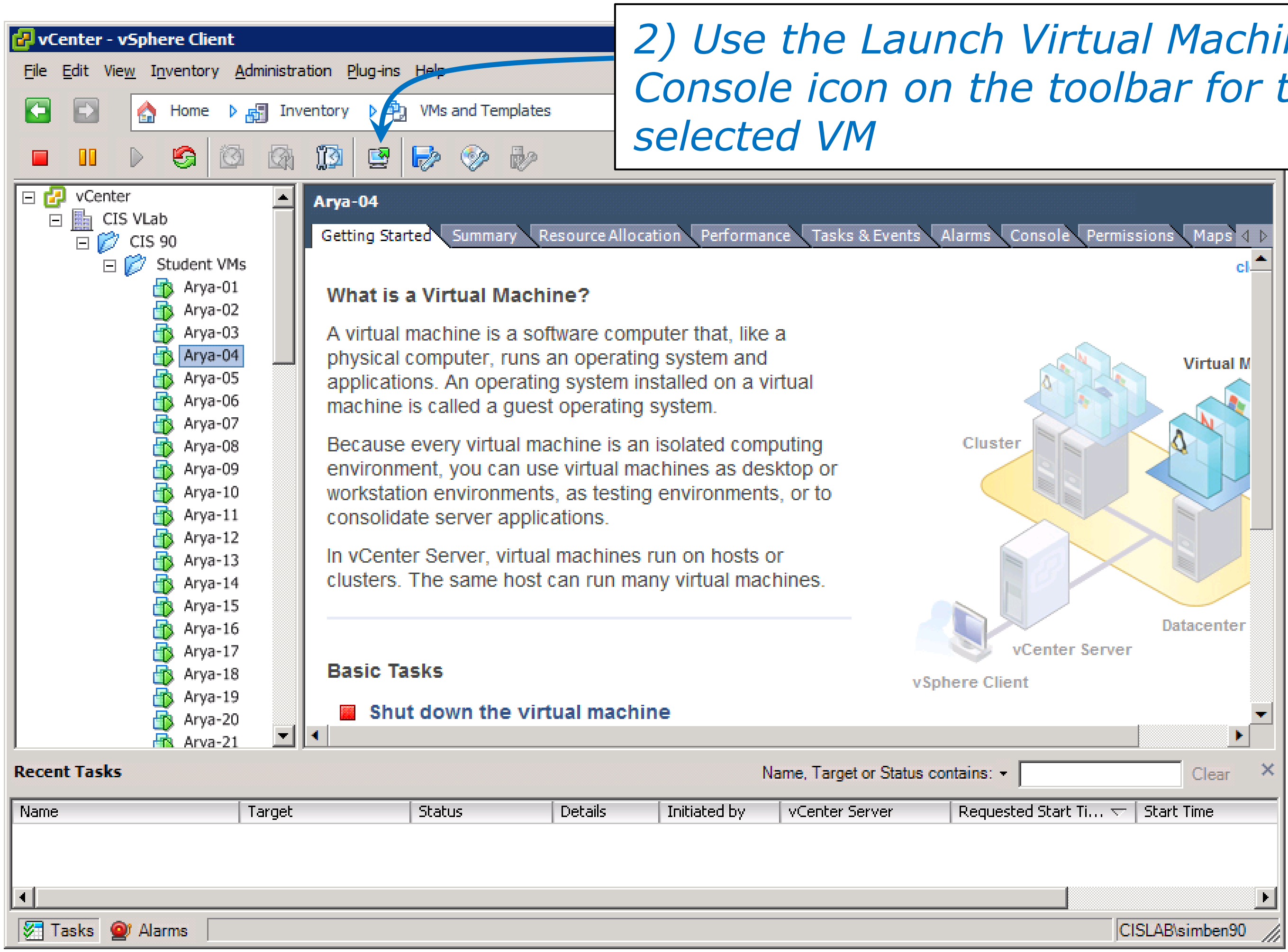
Name	Target	Status	Details	Initiated by	vCenter Server	Requested Start Ti...	Start Time
Initiate guest OS shutd...	Arya-11	Completed		CISLAB\simb...	vCenter	8/24/2014 12:35:17 ...	8/24/2014 12:35:1
Initiate guest OS shutd...	Arya-10	Completed		CISLAB\simb...	vCenter	8/24/2014 12:35:13 ...	8/24/2014 12:35:1

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.

1) Find and select your Arya VM

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

Launching a graphical console



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

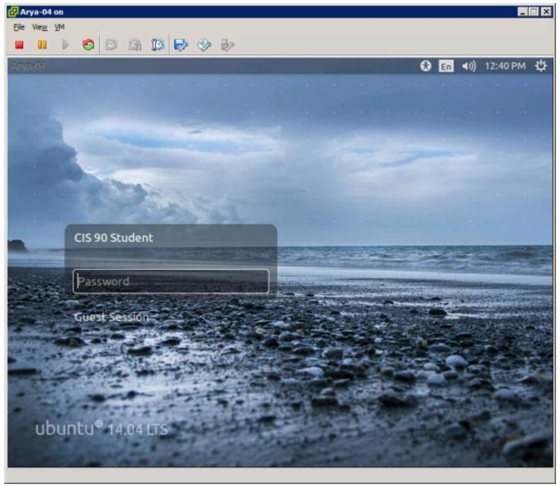
Shut down the virtual machine

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

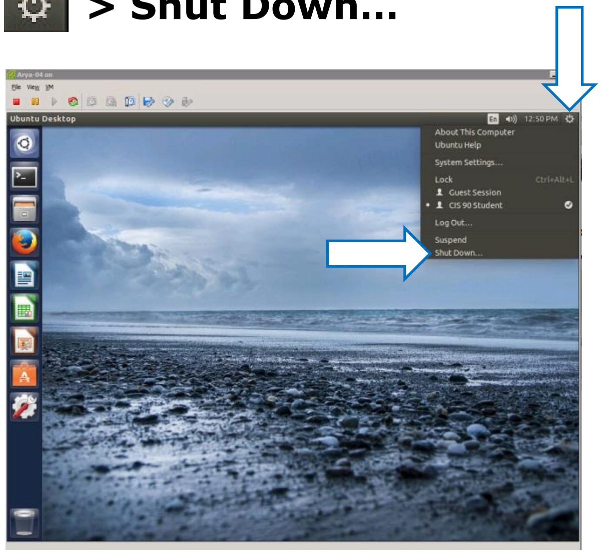
The Arya VM

ubuntu

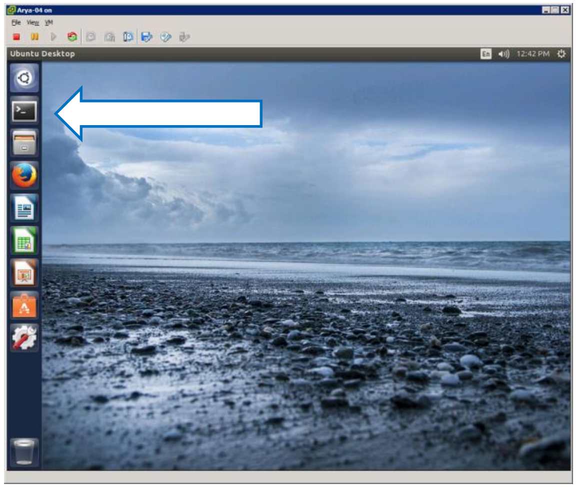
Log in as
CIS 90 Student

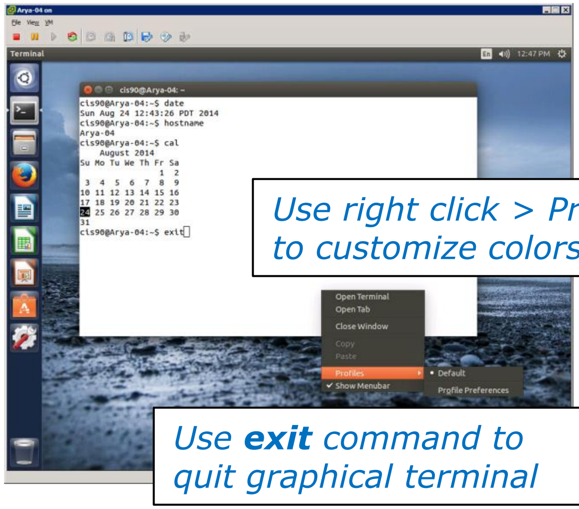


Shutdown using
⚙ **> Shut Down…**



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





```
cis90@Arya-04:~$ date
Sun Aug 24 12:43:26 PDT 2014
cis90@Arya-04:~$ hostname
Arya-04
cis90@Arya-04:~$ cal
    August 2014
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
31
cis90@Arya-04:~$ exit
```

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:

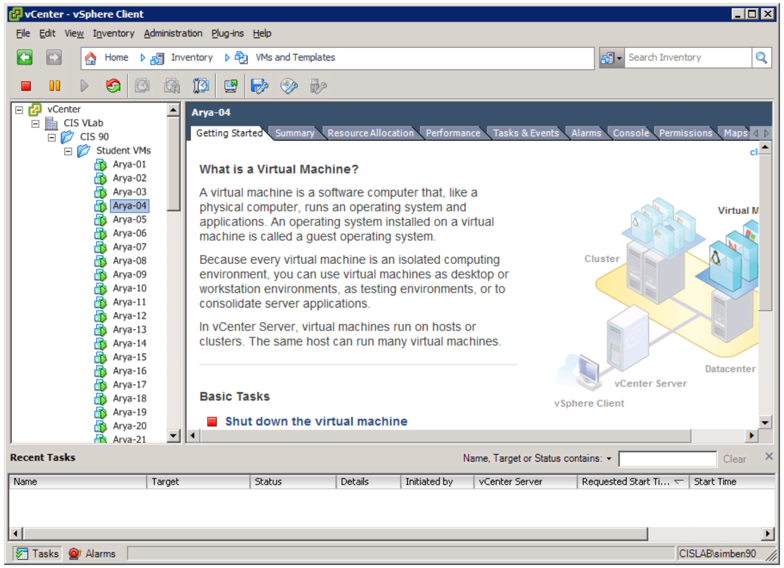
- You just need a command line
- Have a low or high speed network connection
- Note: Windows users can use Putty

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



What is a Virtual Machine?

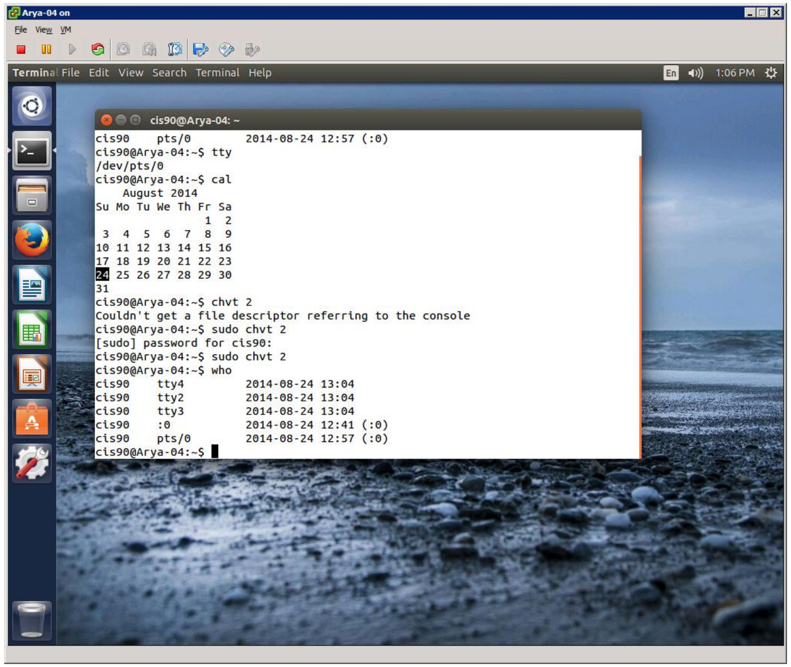
A virtual machine is a software computer that, like a physical computer, runs an operating system and applications. An operating system installed on a virtual machine is called a guest operating system.

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



```
cis90    pts/0        2014-08-24 12:57 (:0)
cis90@Arya-04:~$ tty
/dev/pts/0
cis90@Arya-04:~$ cal
    August 2014
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
31
cis90@Arya-04:~$ chvt 2
Couldn't get a file descriptor referring to the console
cis90@Arya-04:~$ sudo chvt 2
[sudo] password for cis90:
cis90@Arya-04:~$ sudo chvt 2
cis90@Arya-04:~$ who
cis90    tty4         2014-08-24 13:04
cis90    tty2         2014-08-24 13:04
cis90    tty3         2014-08-24 13:04
cis90    :0           2014-08-24 12:41 (:0)
cis90    pts/0        2014-08-24 12:57 (:0)
cis90@Arya-04:~$
```

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

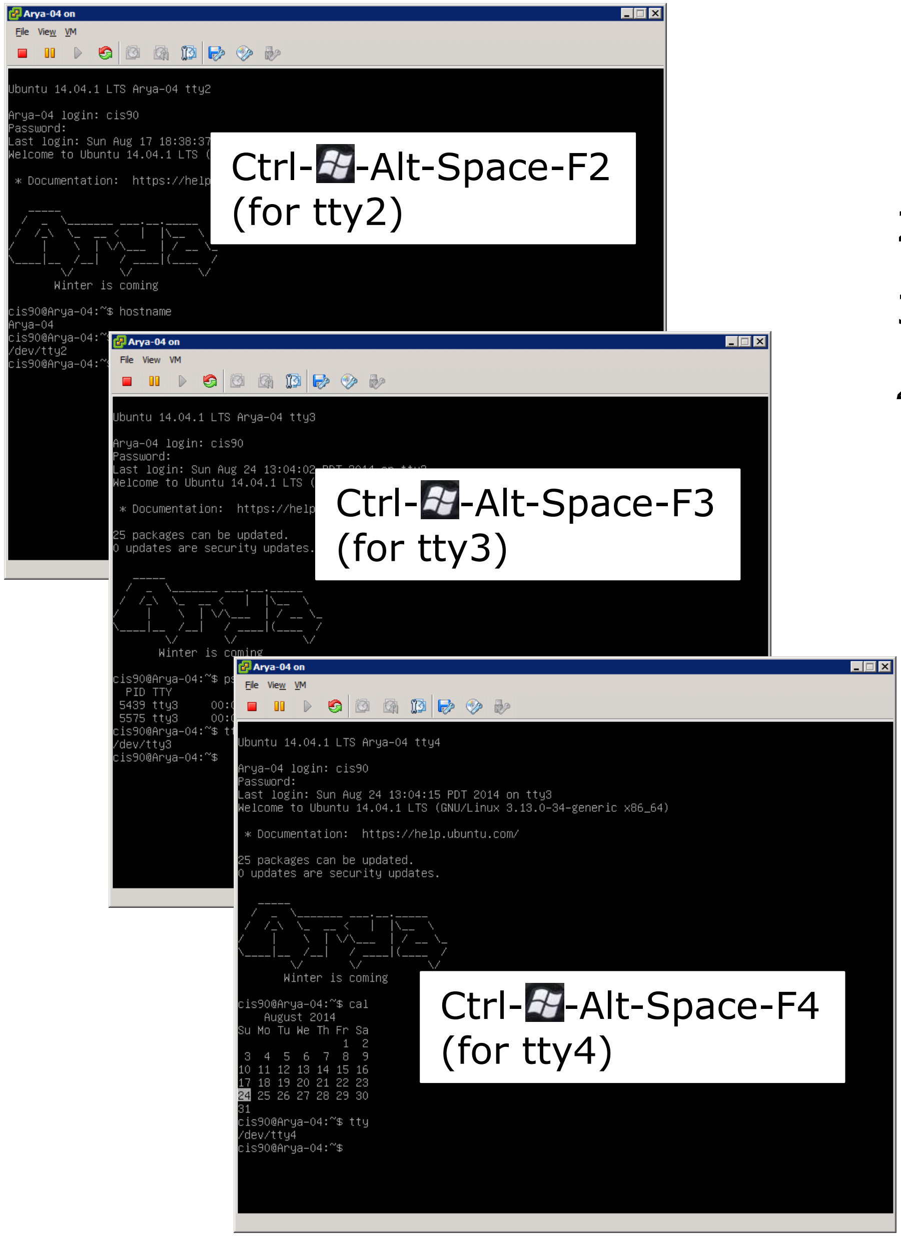
Virtual Terminals

1) While holding down Crtl-⊞-Alt keys, tap Space, then tap F*n* key

2) or try: **chvt** *n*

3) or try: **sudo chvt** *n*

4) or try: <alt-key> n
(in an Ubuntu virtual terminal)



```
Ubuntu 14.04.1 LTS Arya-04 tty2

Arya-04 login: cis90
Password:
Last login: Sun Aug 17 18:38:37
Welcome to Ubuntu 14.04.1 LTS (

 * Documentation:  https://help

  _____
 /  _  \_______ ___.__._____
/  /_\  \_  __ <   |  |\__  \
/    |    \  | \/\___  | / __ \_
\____|__  /__|   / ____|(____  /
        \/       \/          \/
       Winter is coming

cis90@Arya-04:~$ hostname
Arya-04
cis90@Arya-04:~
/dev/tty2
cis90@Arya-04:~
```

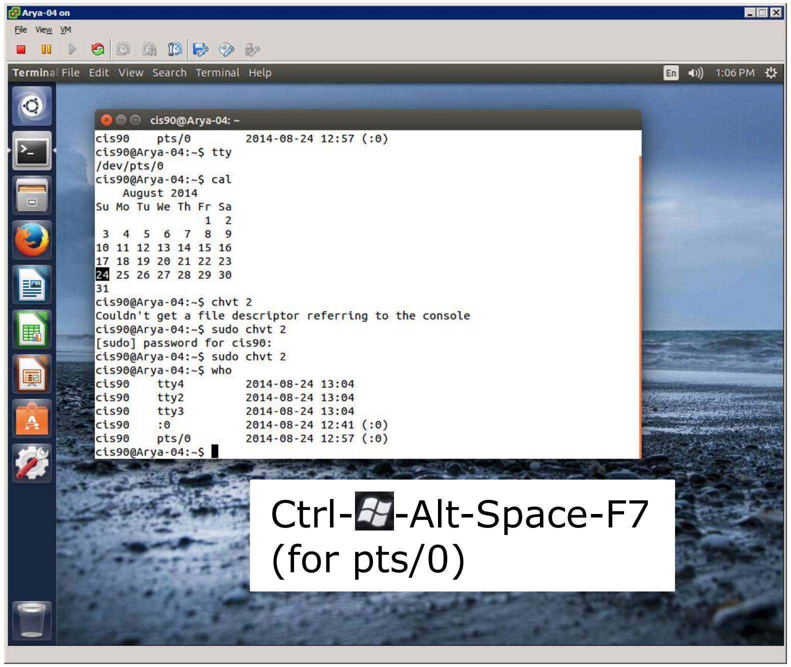
Ctrl-⊞-Alt-Space-F2
(for tty2)

```
Ubuntu 14.04.1 LTS Arya-04 tty3

Arya-04 login: cis90
Password:
Last login: Sun Aug 24 13:04:02
Welcome to Ubuntu 14.04.1 LTS (

 * Documentation:  https://help

25 packages can be updated.
0 updates are security updates.

  _____
 /  _  \_______ ___.__._____
/  /_\  \_  __ <   |  |\__  \
/    |    \  | \/\___  | / __ \_
\____|__  /__|   / ____|(____  /
        \/       \/          \/
       Winter is coming

cis90@Arya-04:~$ ps
  PID TTY
 5439 tty3      00:0
 5575 tty3      00:0
cis90@Arya-04:~$ tt
/dev/tty3
cis90@Arya-04:~$
```

Ctrl-⊞-Alt-Space-F3
(for tty3)

```
Ubuntu 14.04.1 LTS Arya-04 tty4

Arya-04 login: cis90
Password:
Last login: Sun Aug 24 13:04:15 PDT 2014 on tty3
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-34-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

25 packages can be updated.
0 updates are security updates.

  _____
 /  _  \_______ ___.__._____
/  /_\  \_  __ <   |  |\__  \
/    |    \  | \/\___  | / __ \_
\____|__  /__|   / ____|(____  /
        \/       \/          \/
       Winter is coming

cis90@Arya-04:~$ cal
    August 2014
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
31
cis90@Arya-04:~$ tty
/dev/tty4
cis90@Arya-04:~$
```

Ctrl-⊞-Alt-Space-F4
(for tty4)

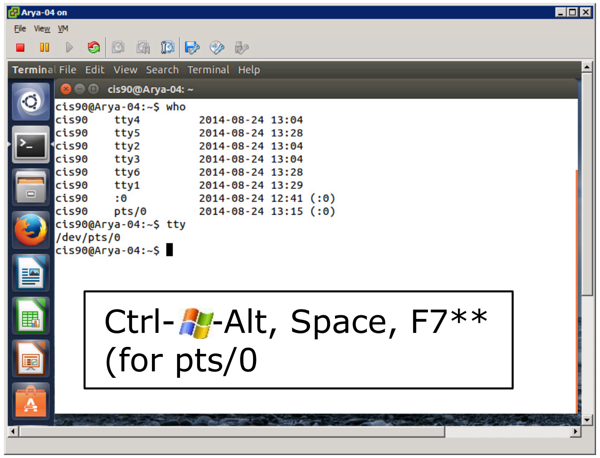


```
cis90    pts/0        2014-08-24 12:57 (:0)
cis90@Arya-04:~$ tty
/dev/pts/0
cis90@Arya-04:~$ cal
    August 2014
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
cis90@Arya-04:~$ chvt 2
Couldn't get a file descriptor referring to the console
cis90@Arya-04:~$ sudo chvt 2
[sudo] password for cis90:
cis90@Arya-04:~$ sudo chvt 2
cis90@Arya-04:~$ who
cis90    tty4         2014-08-24 13:04
cis90    tty2         2014-08-24 13:04
cis90    tty3         2014-08-24 13:04
cis90    :0           2014-08-24 12:41 (:0)
cis90    pts/0        2014-08-24 12:57 (:0)
cis90@Arya-04:~$
```

Ctrl-⊞-Alt-Space-F7
(for pts/0)

Changing Virtual TTY Terminals using **VMware vSphere**

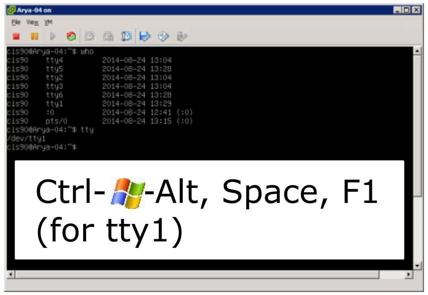
Windows PC Keyboard



```
cis90@Arya-04:~$ who
cis90    tty4         2014-08-24 13:04
cis90    tty5         2014-08-24 13:28
cis90    tty2         2014-08-24 13:04
cis90    tty3         2014-08-24 13:04
cis90    tty6         2014-08-24 13:28
cis90    tty1         2014-08-24 13:29
cis90    :0           2014-08-24 12:41 (:0)
cis90    pts/0        2014-08-24 13:15 (:0)
cis90@Arya-04:~$ tty
/dev/pts/0
cis90@Arya-04:~$ █
```

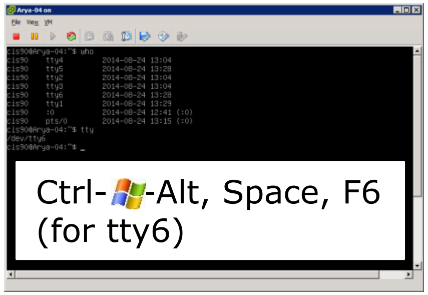
Ctrl-⊞-Alt, Space, F7**
(for pts/0

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



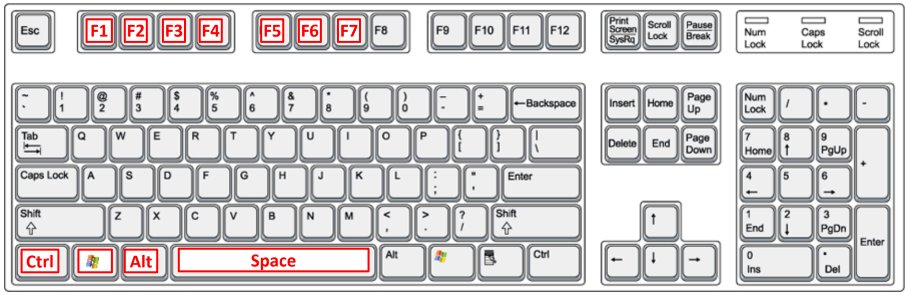
```
cis90@Arya-04:~$ who
cis90    tty4         2014-08-24 13:04
cis90    tty5         2014-08-24 13:28
cis90    tty2         2014-08-24 13:04
cis90    tty3         2014-08-24 13:04
cis90    tty6         2014-08-24 13:28
cis90    tty1         2014-08-24 13:29
cis90    :0           2014-08-24 12:41 (:0)
cis90    pts/0        2014-08-24 13:15 (:0)
cis90@Arya-04:~$ tty
/dev/tty1
cis90@Arya-04:~$
```

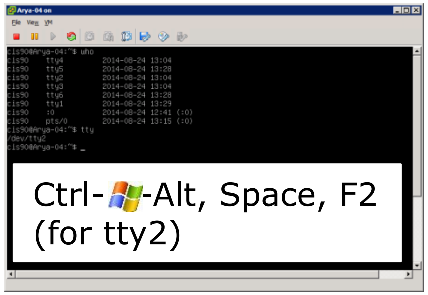
Ctrl-⊞-Alt, Space, F1
(for tty1)



```
cis90@Arya-04:~$ who
cis90    tty4         2014-08-24 13:04
cis90    tty5         2014-08-24 13:28
cis90    tty2         2014-08-24 13:04
cis90    tty3         2014-08-24 13:04
cis90    tty6         2014-08-24 13:28
cis90    tty1         2014-08-24 13:29
cis90    :0           2014-08-24 12:41 (:0)
cis90    pts/0        2014-08-24 13:15 (:0)
cis90@Arya-04:~$ tty
/dev/tty6
cis90@Arya-04:~$ _
```

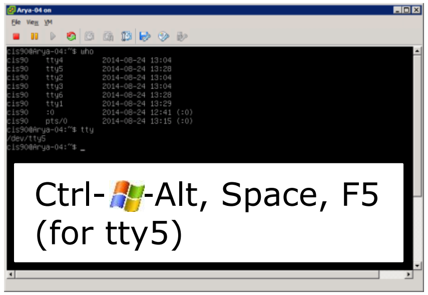
Ctrl-⊞-Alt, Space, F6
(for tty6)





```
cis90@Arya-04:~$ who
cis90    tty4         2014-08-24 13:04
cis90    tty5         2014-08-24 13:28
cis90    tty2         2014-08-24 13:04
cis90    tty3         2014-08-24 13:04
cis90    tty6         2014-08-24 13:28
cis90    tty1         2014-08-24 13:29
cis90    :0           2014-08-24 12:41 (:0)
cis90    pts/0        2014-08-24 13:15 (:0)
cis90@Arya-04:~$ tty
/dev/tty2
cis90@Arya-04:~$ _
```

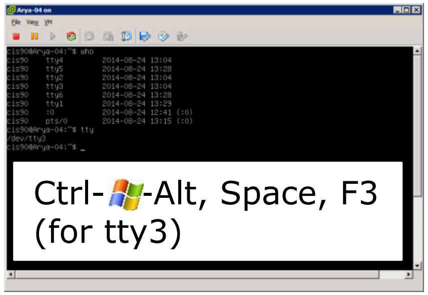
Ctrl-⊞-Alt, Space, F2
(for tty2)



```
cis90@Arya-04:~$ who
cis90    tty4         2014-08-24 13:04
cis90    tty5         2014-08-24 13:28
cis90    tty2         2014-08-24 13:04
cis90    tty3         2014-08-24 13:04
cis90    tty6         2014-08-24 13:28
cis90    tty1         2014-08-24 13:29
cis90    :0           2014-08-24 12:41 (:0)
cis90    pts/0        2014-08-24 13:15 (:0)
cis90@Arya-04:~$ tty
/dev/tty5
cis90@Arya-04:~$ _
```

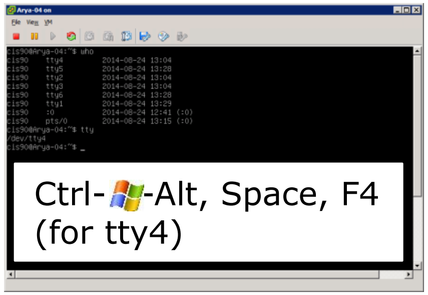
Ctrl-⊞-Alt, Space, F5
(for tty5)

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



```
cis90@Arya-04:~$ who
cis90    tty4         2014-08-24 13:04
cis90    tty5         2014-08-24 13:28
cis90    tty2         2014-08-24 13:04
cis90    tty3         2014-08-24 13:04
cis90    tty6         2014-08-24 13:28
cis90    tty1         2014-08-24 13:29
cis90    :0           2014-08-24 12:41 (:0)
cis90    pts/0        2014-08-24 13:15 (:0)
cis90@Arya-04:~$ tty
/dev/tty3
cis90@Arya-04:~$ _
```

Ctrl-⊞-Alt, Space, F3
(for tty3)



```
cis90@Arya-04:~$ who
cis90    tty4         2014-08-24 13:04
cis90    tty5         2014-08-24 13:28
cis90    tty2         2014-08-24 13:04
cis90    tty3         2014-08-24 13:04
cis90    tty6         2014-08-24 13:28
cis90    tty1         2014-08-24 13:29
cis90    :0           2014-08-24 12:41 (:0)
cis90    pts/0        2014-08-24 13:15 (:0)
cis90@Arya-04:~$ tty
/dev/tty4
cis90@Arya-04:~$ _
```

Ctrl-⊞-Alt, Space, F4
(for tty4)

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

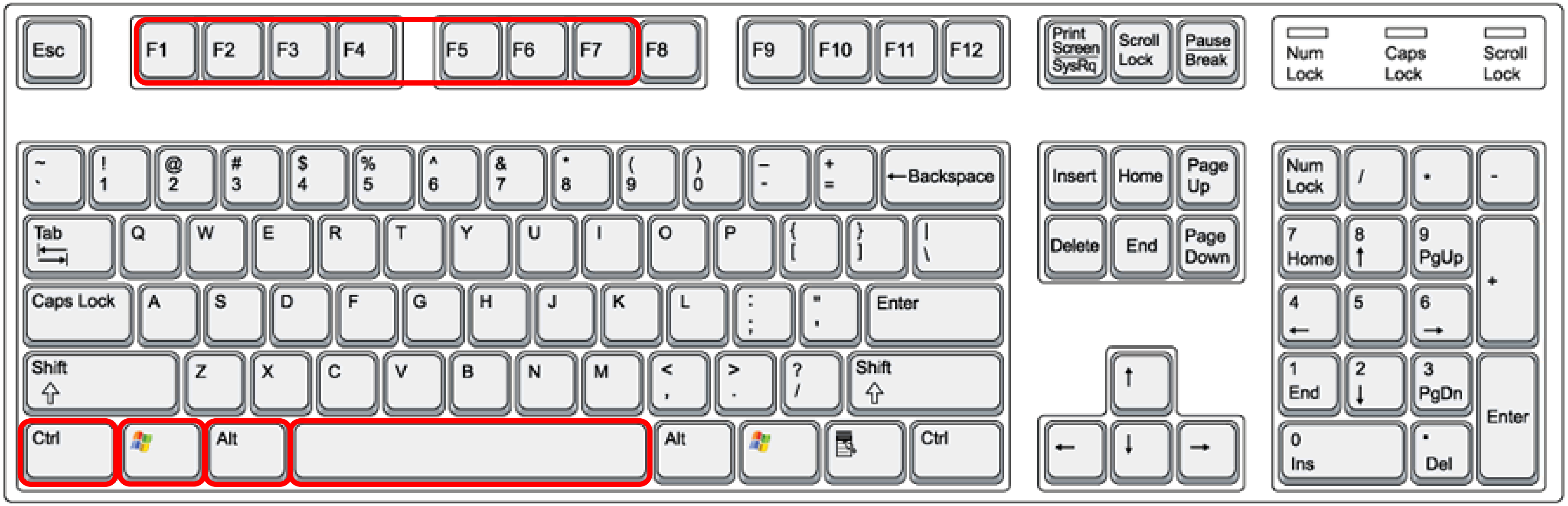
F7 is graphics mode for the Ubuntu VMs.

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

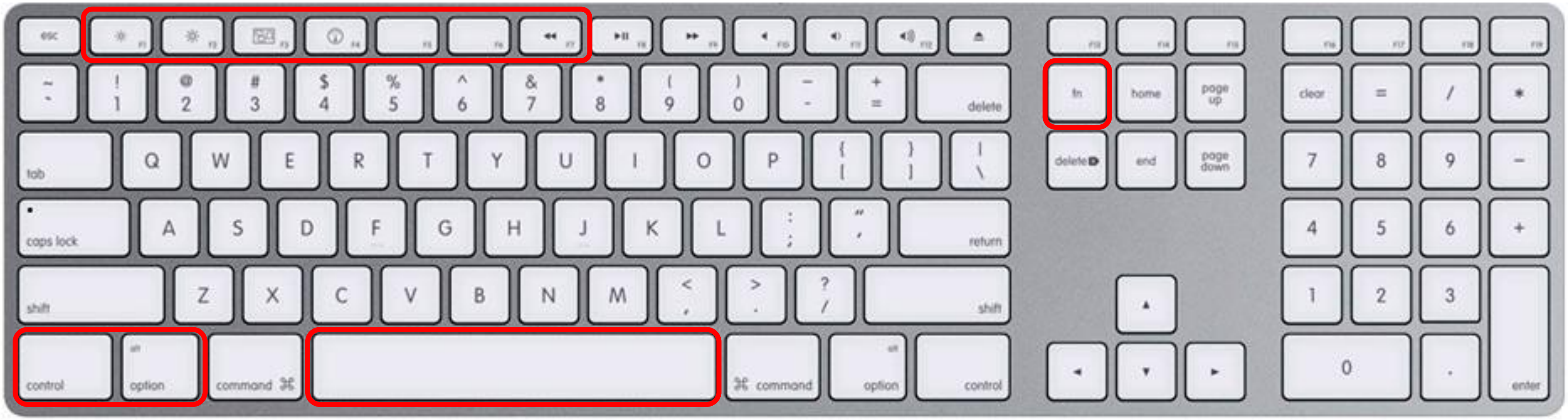

Esc
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
Print Screen SysRq
Scroll Lock
Pause Break
Num Lock
Caps Lock
Scroll Lock
~ ` ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 _ - + = ←Backspace
Insert Home Page Up
Num Lock / * -
Tab Q W E R T Y U I O P { [}] | \
Delete End Page Down
7 Home 8 ↑ 9 PgUp +
Caps Lock A S D F G H J K L : ; " ' Enter
4 ← 5 6 →
Shift Z X C V B N M < , > . ? / Shift
↑
1 End 2 ↓ 3 PgDn Enter
Ctrl Alt Alt Ctrl
← ↓ →
0 Ins . Del

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

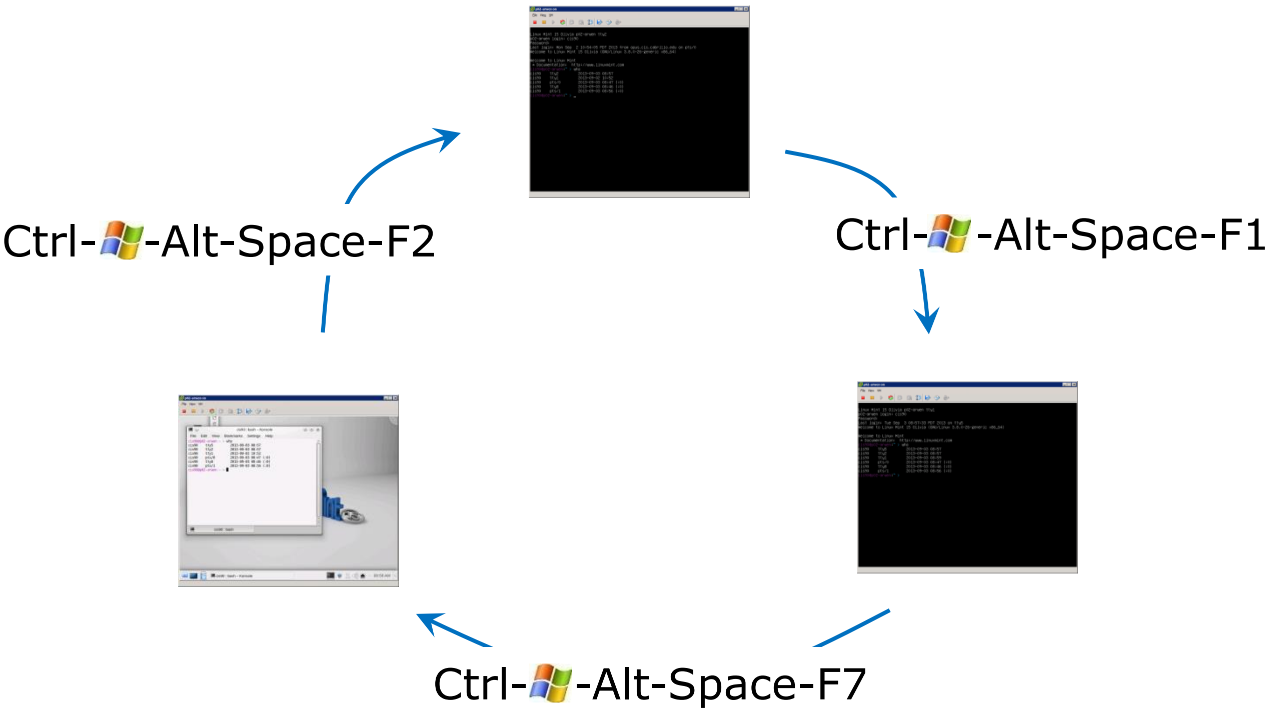
Changing Virtual Terminals with a Mac keyboard



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



Ctrl-⊞-Alt-Space-F2

Ctrl-⊞-Alt-Space-F1

Ctrl-⊞-Alt-Space-F7

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
CORPORATION
SUBSIDIARY OF
Western Electric Company
CHICAGO, U.S.A.

Teletype (TTY)

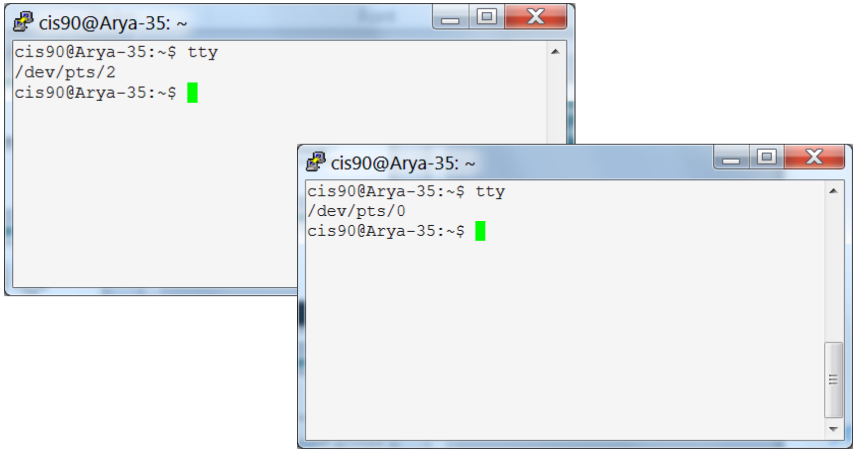
VT100

digital

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

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

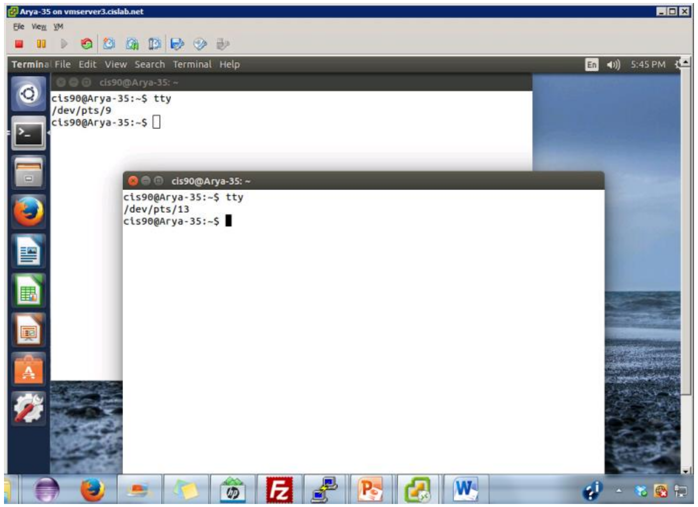
Software Terminals



```
cis90@Arya-35:~$ tty
/dev/pts/2
cis90@Arya-35:~$
```

```
cis90@Arya-35:~$ tty
/dev/pts/0
cis90@Arya-35:~$
```

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



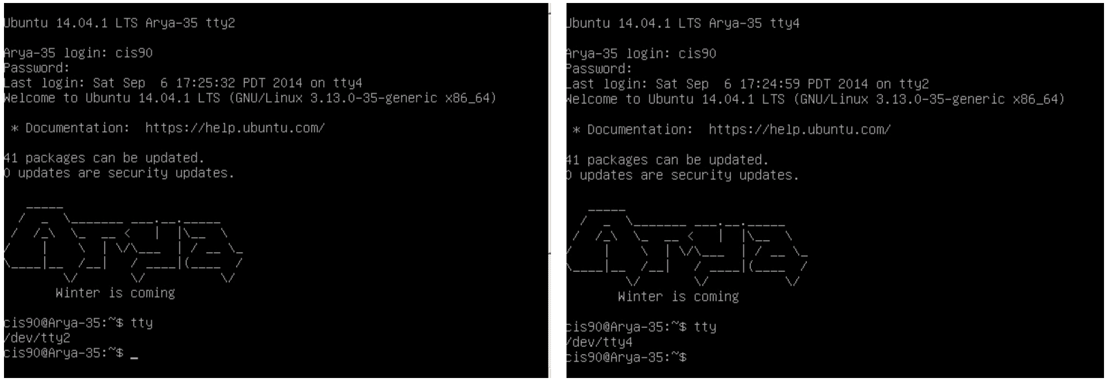
Arya-35 on vmserver3.cislab.net
File View VM
Terminal File Edit View Search Terminal Help
En 5:45 PM
cis90@Arya-35: ~

```
cis90@Arya-35:~$ tty
/dev/pts/9
cis90@Arya-35:~$
```

```
cis90@Arya-35:~$ tty
/dev/pts/13
cis90@Arya-35:~$
```

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

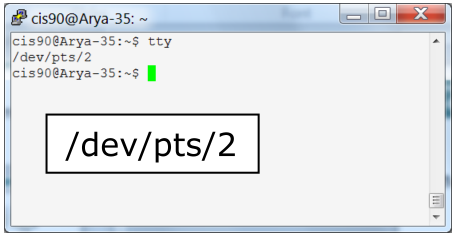


```
Ubuntu 14.04.1 LTS Arya-35 tty2

Arya-35 login: cis90
Password:
Last login: Sat Sep  6 17:25:32 PDT 2014 on tty4
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-35-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

41 packages can be updated.
0 updates are security updates.

   _____
  /  _  \_______ ___.__._____
 /  /_\  \_  __ <   |  |\__  \
/    |    \  | \/\___  | / __ \_
\____|__  /__|   / ____|(____  /
        \/       \/          \/
       Winter is coming

cis90@Arya-35:~$ tty
/dev/tty2
cis90@Arya-35:~$ _
```

```
Ubuntu 14.04.1 LTS Arya-35 tty4

Arya-35 login: cis90
Password:
Last login: Sat Sep  6 17:24:59 PDT 2014 on tty2
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-35-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

41 packages can be updated.
0 updates are security updates.

   _____
  /  _  \_______ ___.__._____
 /  /_\  \_  __ <   |  |\__  \
/    |    \  | \/\___  | / __ \_
\____|__  /__|   / ____|(____  /
        \/       \/          \/
       Winter is coming

cis90@Arya-35:~$ tty
/dev/tty4
cis90@Arya-35:~$
```

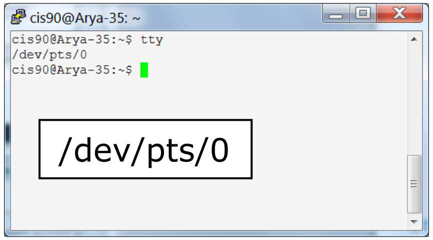
Various terminal devices on an Arya VM

Terminal emulators (e.g. Putty)



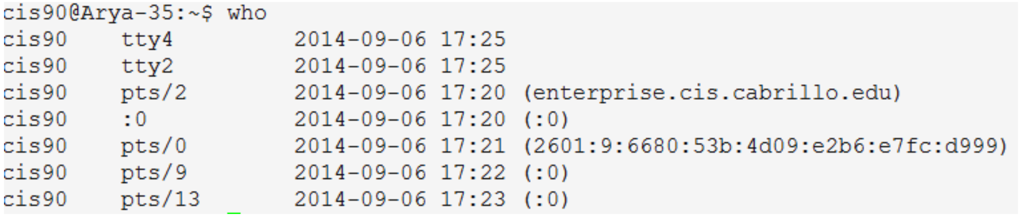
```
cis90@Arya-35:~$ tty
/dev/pts/2
cis90@Arya-35:~$
```

/dev/pts/2



```
cis90@Arya-35:~$ tty
/dev/pts/0
cis90@Arya-35:~$
```

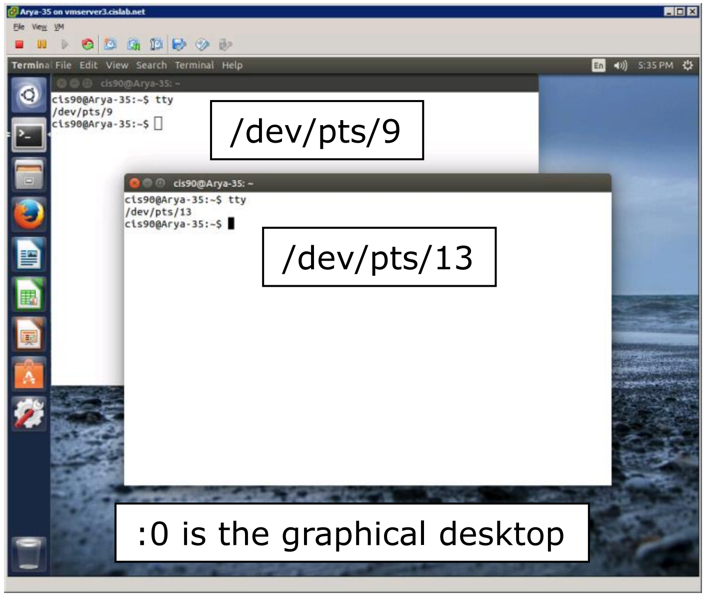
/dev/pts/0



```
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


Arya-35 on vmserver3.cislab.net
File View VM
Termina File Edit View Search Terminal Help
En
5:35 PM
cis90@Arya-35: ~
cis90@Arya-35: ~

```
cis90@Arya-35:~$ tty
/dev/pts/9
cis90@Arya-35:~$
```

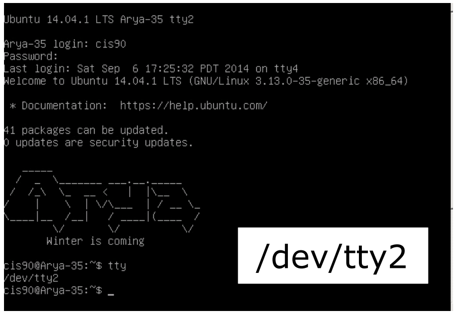
/dev/pts/9

```
cis90@Arya-35:~$ tty
/dev/pts/13
cis90@Arya-35:~$
```

/dev/pts/13

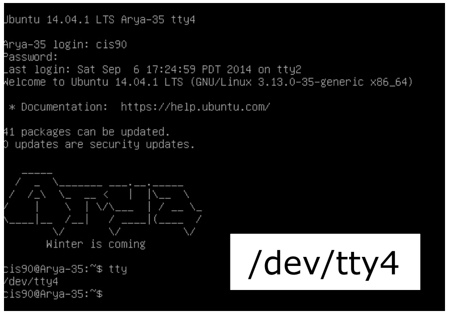
:0 is the graphical desktop

Virtual terminals



```
Ubuntu 14.04.1 LTS Arya-35 tty2

Arya-35 login: cis90
Password:
Last login: Sat Sep  6 17:25:32 PDT 2014 on tty4
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-35-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

41 packages can be updated.
0 updates are security updates.

   _____
  /  _  \_______ ___.__.______
 /  /_\  \_  __ <   |  |\__  \
/    |    \  | \/\___  | / __ \_
\____|__  /__|   / ____|(____  /
        \/       \/          \/
       Winter is coming

cis90@Arya-35:~$ tty
/dev/tty2
cis90@Arya-35:~$ _
```

/dev/tty2



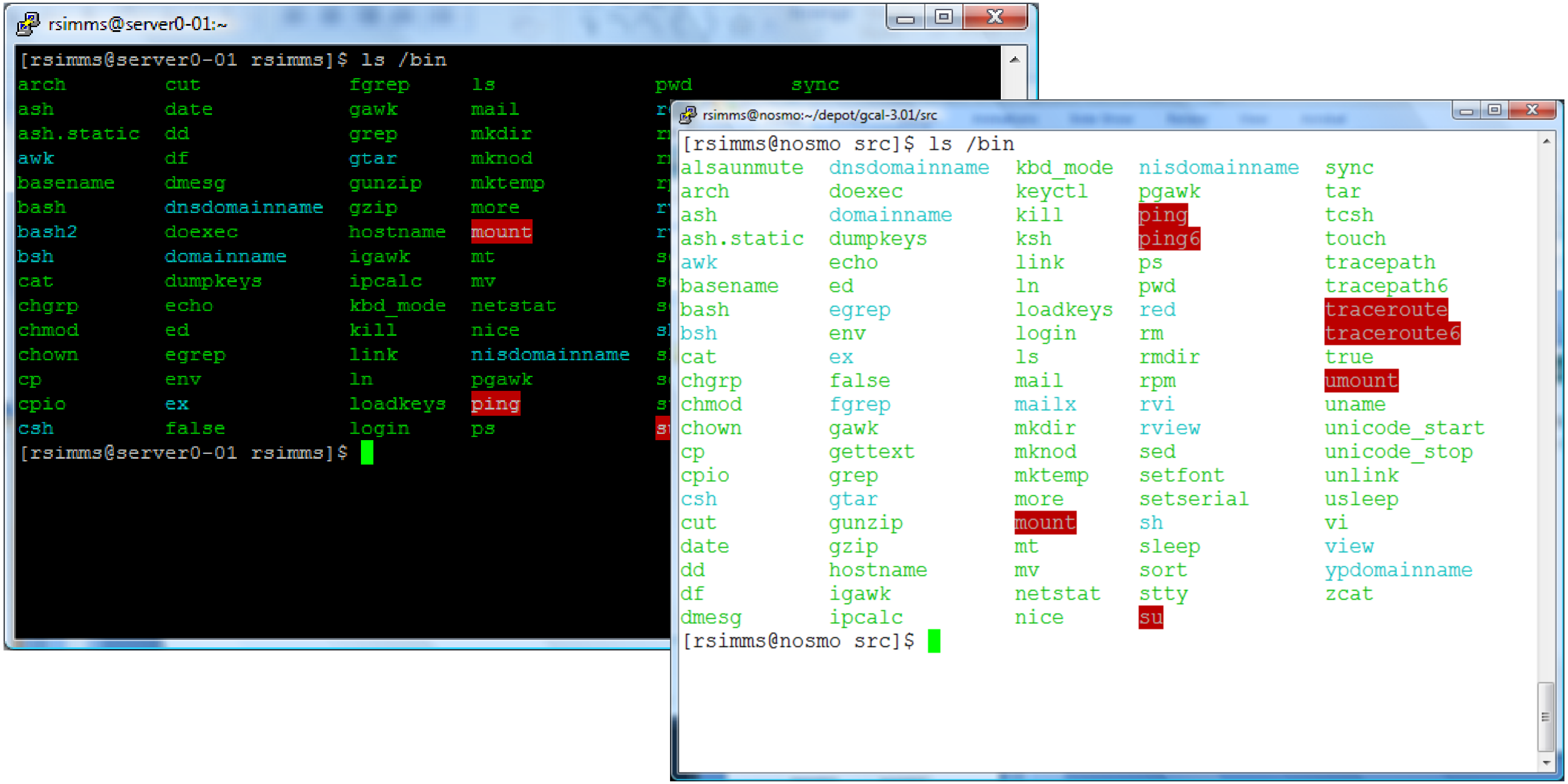
```
Ubuntu 14.04.1 LTS Arya-35 tty4

Arya-35 login: cis90
Password:
Last login: Sat Sep  6 17:24:59 PDT 2014 on tty2
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-35-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

41 packages can be updated.
0 updates are security updates.

   _____
  /  _  \_______ ___.__.______
 /  /_\  \_  __ <   |  |\__  \
/    |    \  | \/\___  | / __ \_
\____|__  /__|   / ____|(____  /
        \/       \/          \/
       Winter is coming

cis90@Arya-35:~$ tty
/dev/tty4
cis90@Arya-35:~$
```

/dev/tty4

Putty
Tips

(Note: tty = teletype)

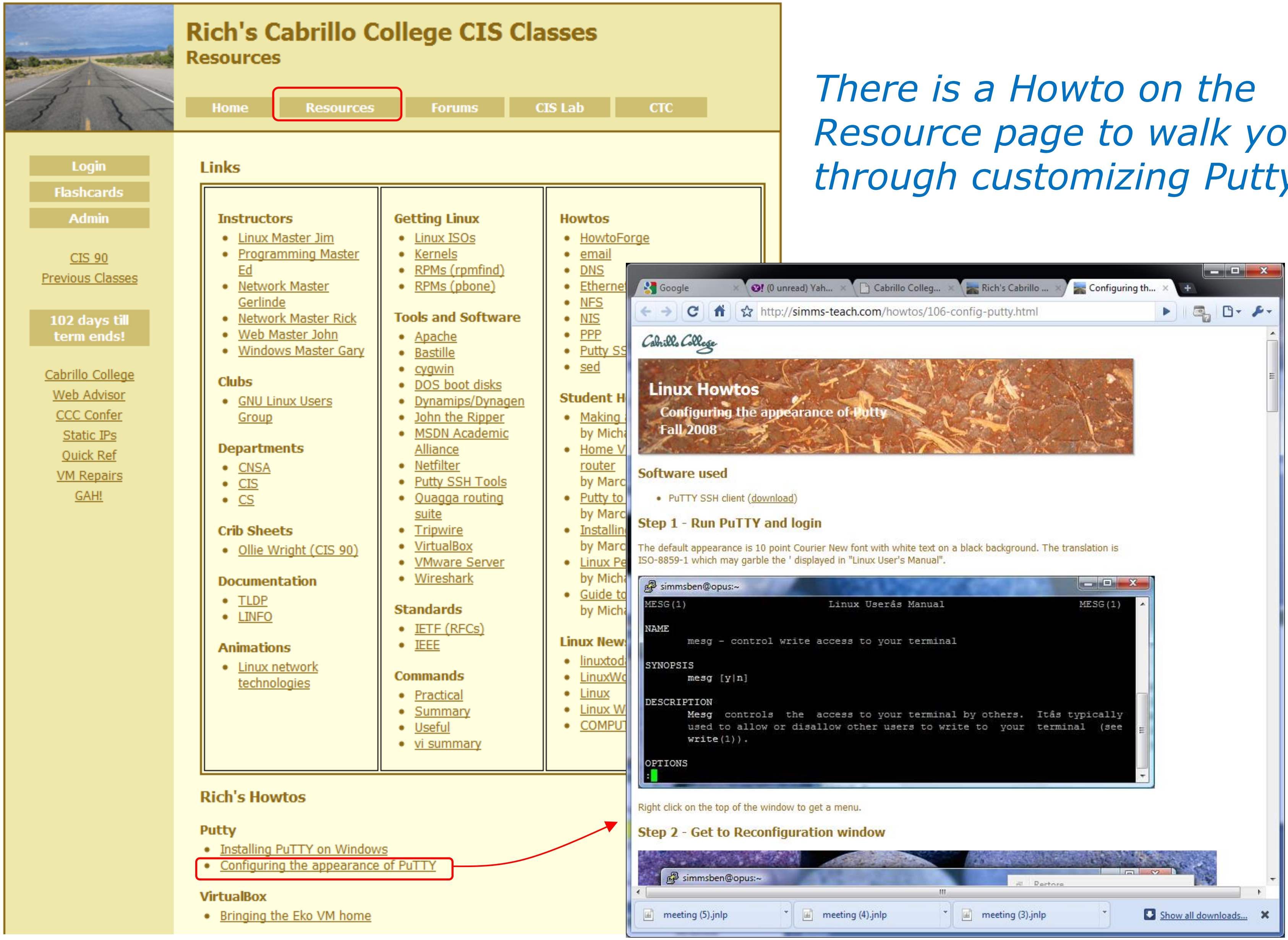
The Putty program



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

```
[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

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
- 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

Howtos
- HowtoForge
- email
- DNS
- Etherne
- NFS
- NIS
- PPP
- Putty SS
- sed

Student H
- Making by Micha
- Home V router by Marc
- Putty to by Marc
- Installin by Marc
- Linux Pe by Micha
- Guide to by Micha

Linux New
- linuxtod
- LinuxWo
- Linux
- Linux W
- COMPU

Rich's Howtos

Putty
- Installing PuTTY on Windows
- Configuring the appearance of PuTTY

VirtualBox
- Bringing the Eko VM home

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".

```
MESG(1)                     Linux Userâs Manual                      MESG(1)

NAME
       mesg - control write access to your terminal

SYNOPSIS
       mesg [y|n]

DESCRIPTION
       Mesg  controls  the  access to your terminal by others.  Itâs typically
       used to allow or disallow other users to write to  your  terminal  (see
       write(1)).

OPTIONS
:
```

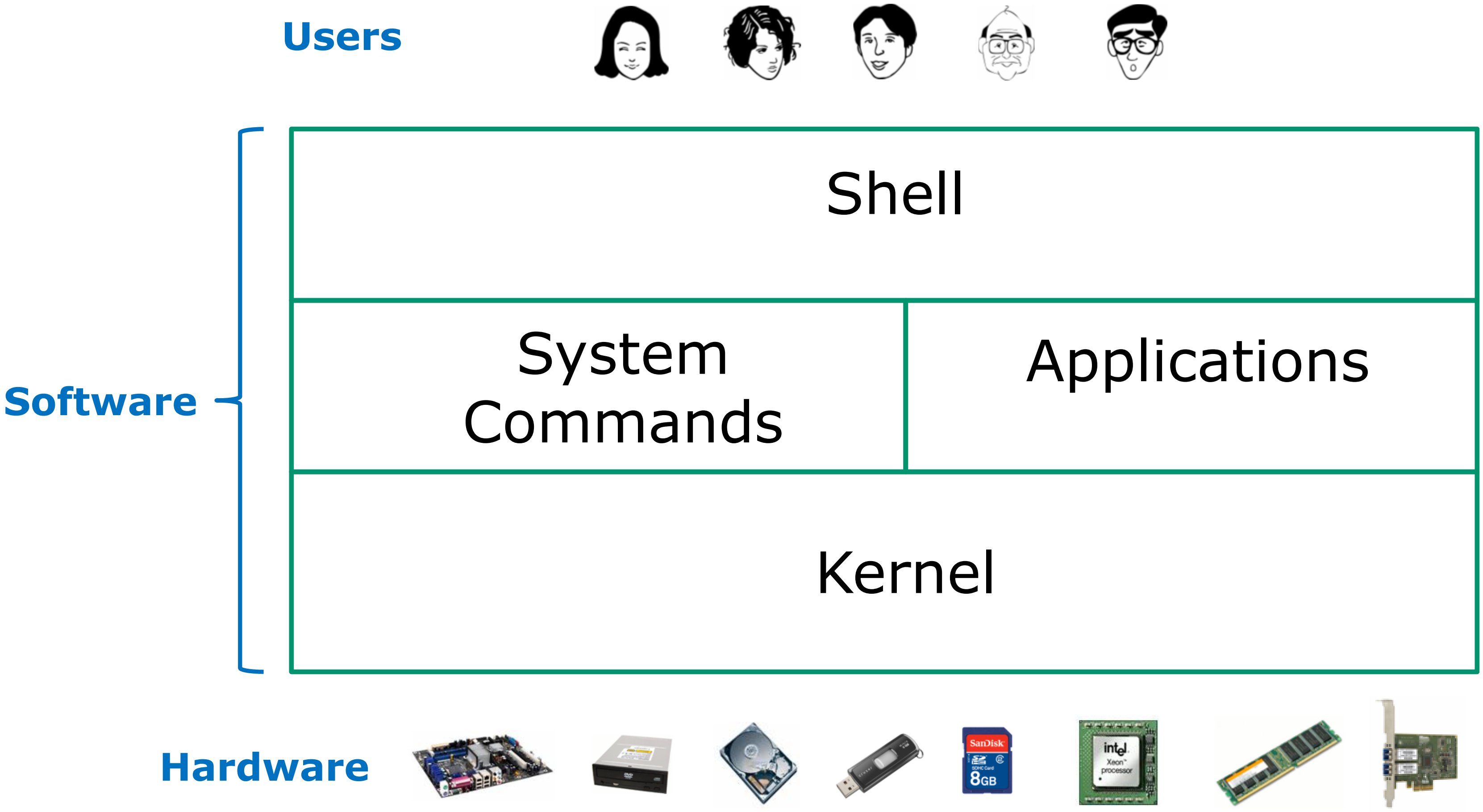
Right click on the top of the window to get a menu.

Step 2 - Get to Reconfiguration window

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

Lesson 1 Review

UNIX/Linux Architecture
Simplified View - Four Major Components



Users

Software

Shell

System Commands

Applications

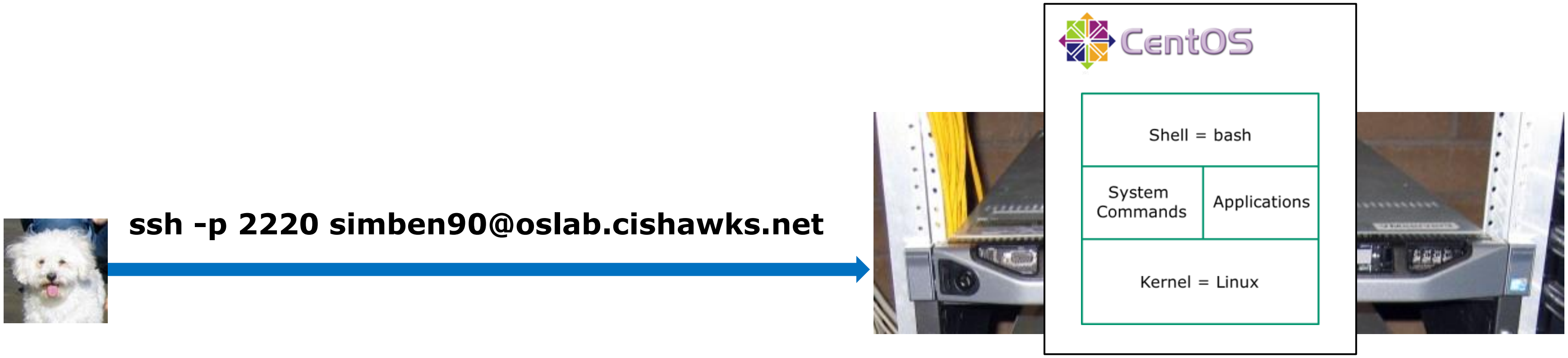
Kernel

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

CentOS

Shell = bash

System Commands

Applications

Kernel = Linux

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

Various "names" bandied about:	**To view:**
User's first and last **name**: Benji Simms	*/etc/passwd*
user**name** = simben90	**id**
name of terminal device used = /dev/pts/2	**tty**
(terminal type = xterm)	**echo $TERM**
host**name** = oslab.cishawks.net	**hostname**
Name of distro = CentOS	*/etc/issue*
Name of shell = bash	**ps**
Name of kernel = Linux	**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')
                                //-=-\\
                                (\_=_/)
                                 ~~ ~~
                             Welcome to Opus
                        Serving Cabrillo College

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

Hit Enter to accept

*The terminal <u>type</u> is **xterm***

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

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

How can I print a calendar?

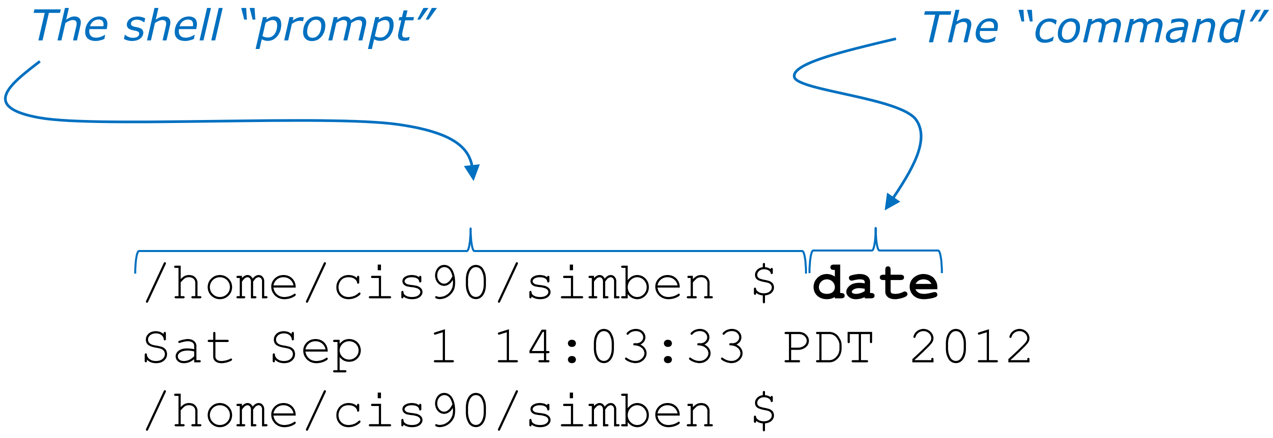
```
/home/cis90/simben $ cal
   September 2012
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30

/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
/home/cis90/simben $
```

*The **cal** command*

*Month and year **arguments***

A command can have arguments

What is the current time and date?



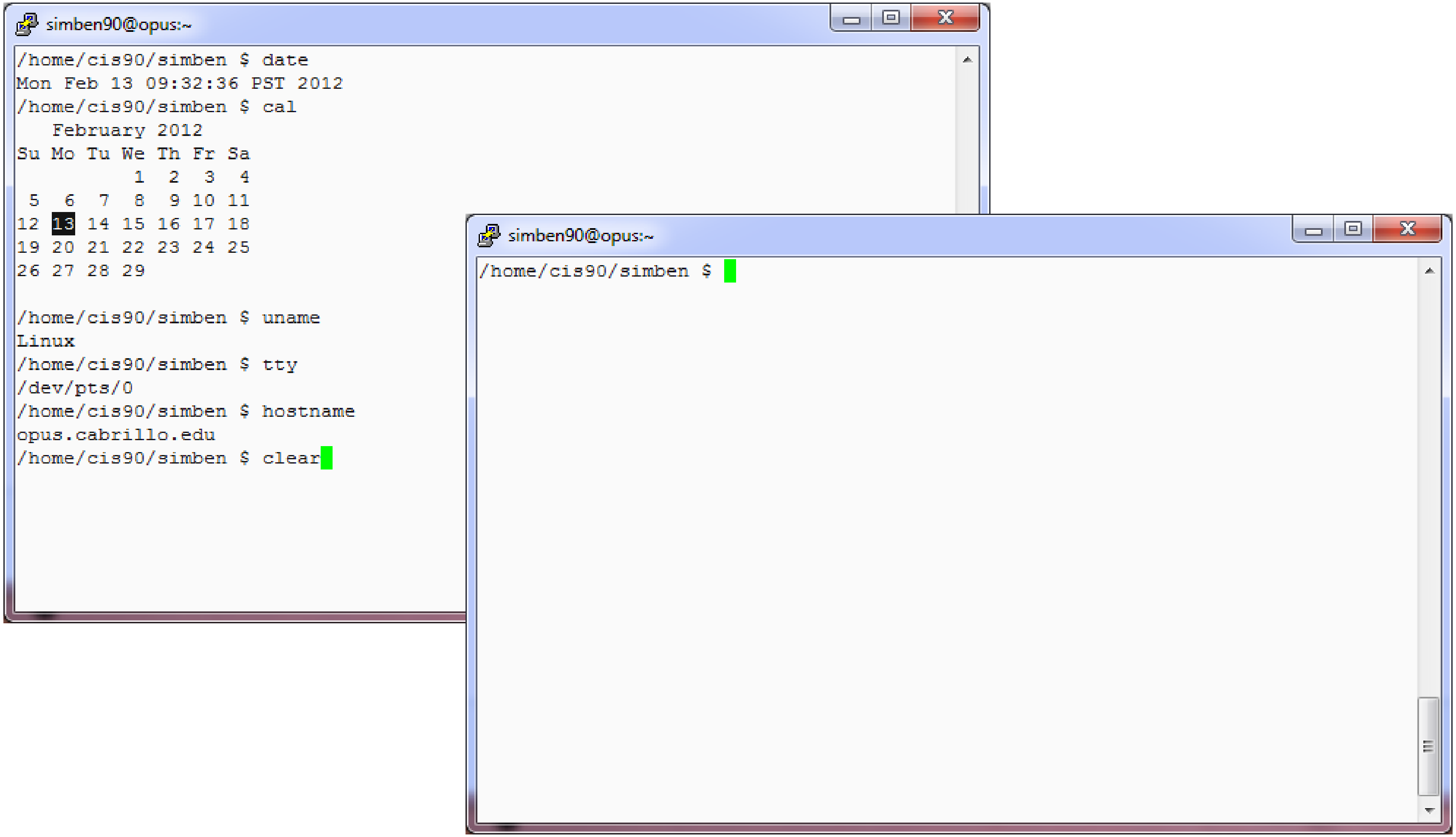
The shell "prompt"

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?



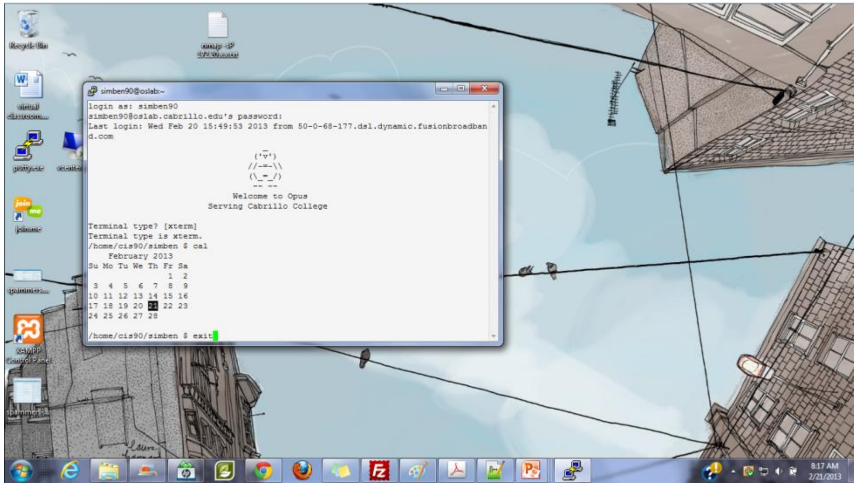
```
/home/cis90/simben $ date
Mon Feb 13 09:32:36 PST 2012
/home/cis90/simben $ cal
   February 2012
Su Mo Tu We Th Fr Sa
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29

/home/cis90/simben $ uname
Linux
/home/cis90/simben $ tty
/dev/pts/0
/home/cis90/simben $ hostname
opus.cabrillo.edu
/home/cis90/simben $ clear
```

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

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

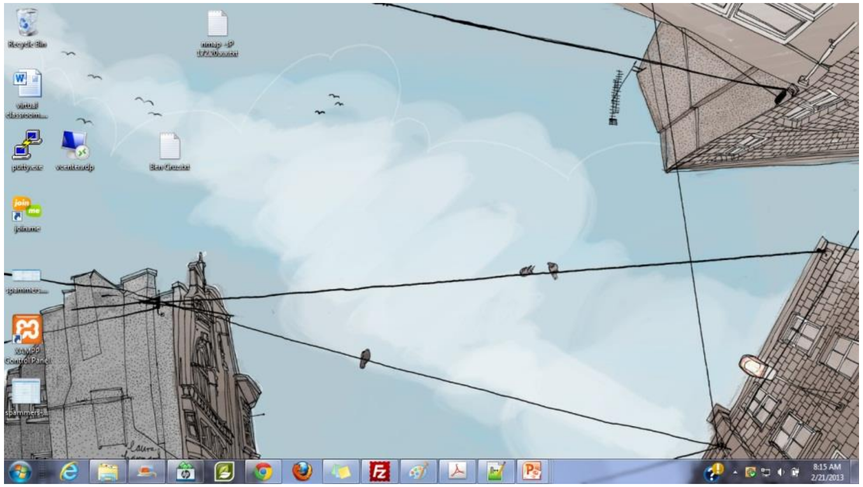
How do I end this login session?

before **exit**



```
login as: simben90
simben90@oslab.cabrillo.edu's password:
Last login: Wed Feb 20 15:49:53 2013 from 50-0-68-177.dsl.dynamic.fusionbroadban
d.com

                                   _
                                 ('v')
                                //-=-\\
                                (\_=_/)
                                 ~~ ~~
                             Welcome to Opus
                        Serving Cabrillo College

Terminal type? [xterm]
Terminal type is xterm.
/home/cis90/simben $ cal
    February 2013
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

/home/cis90/simben $ 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?

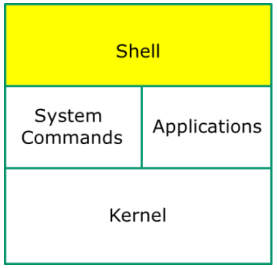
```
/home/cis90/simben $ id
uid=1001(simben90) gid=190(cis90) groups=190(cis90),100(users)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

/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)
```

Usernames

UID's (user ID numbers)

We are all just numbers to the Linux kernel

What shell am I using?



```
/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

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

Terminal device being used

***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 am IP address*

username on remote computer

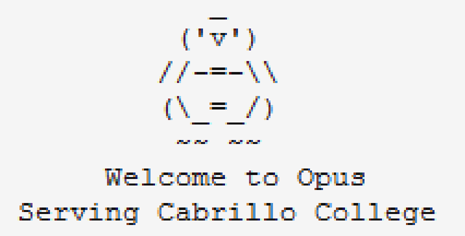
IP address of remote computer

```
/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)

 * 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:~$
```

Notice how the prompt changes on the remote computer

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

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



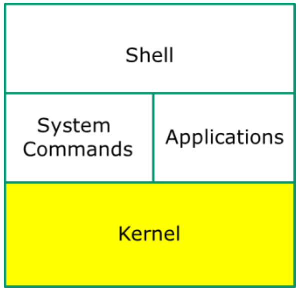
```
     _
   ('v')
  //-=-\\
  (\_=_/)
   ~~ ~~
 Welcome to Opus
Serving Cabrillo College
```

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.

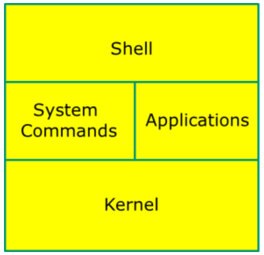
What kernel am I running on?



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

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

What "distro" has been installed?



Shell

System Commands

Applications

Kernel

```
/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)
```

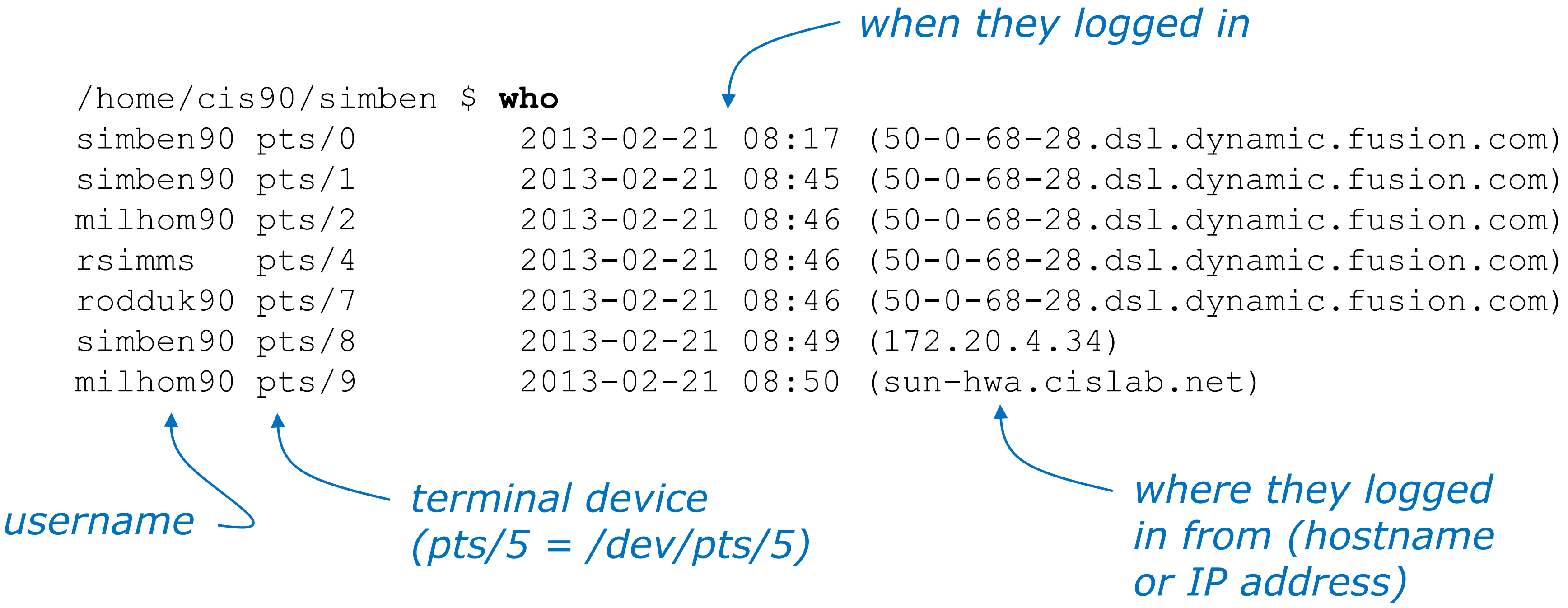
Catting out these files *usually* will show the distro name

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)
```

when they logged in

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.