



Rich's lesson module checklist

Last Modified 2/6/2017

- ☐ Slides and lab posted
- ☐ WB converted from PowerPoint
- ☐ Print out agenda slide and annotate page numbers

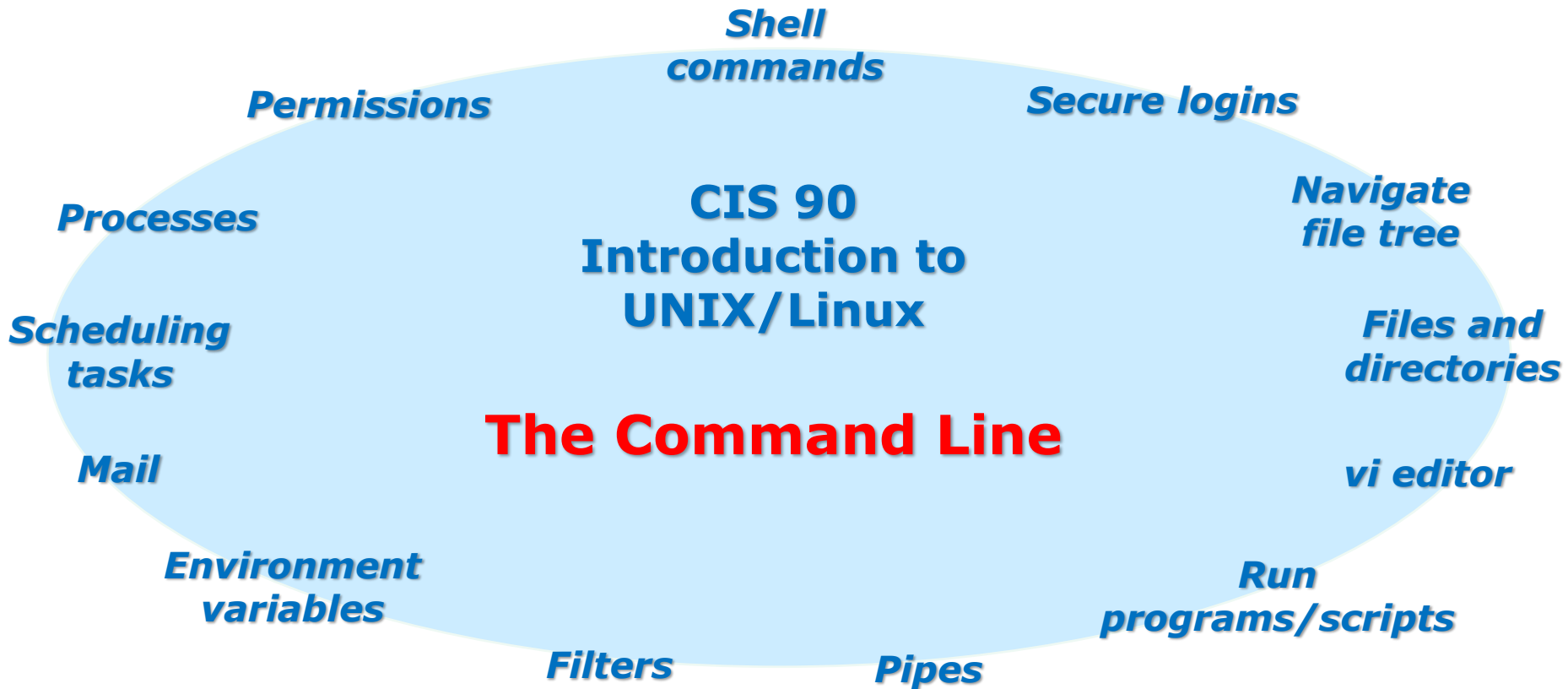
- ☐ Flash cards
- ☐ Properties
- ☐ Page numbers
- ☐ 1st minute quiz
- ☐ Web Calendar summary
- ☐ Web book pages
- ☐ Commands

- ☐ Lab 2 tested (check Q11 kernel release number and finger user account)
- ☐ Opus – set submit deadline
 - at 12:00 am lateday
 - chmod 700 /home/cis90/bin/submit
 - chmod 700 /home/turnin/cis90
 - at 9:00 am lateday
 - chmod 750 /home/cis90/bin/submit
 - chmod 755 /home/turnin/cis90

- ☐ Bring Add Codes
- ☐ Bring printed roster

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

- ☐ Update CCC Confer and 3C Media portals



Student Learner Outcomes

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

Introductions and Credits



Jim Griffin

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



Rich Simms

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

And thanks to:

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



Student checklist for attending class

The screenshot shows a web browser window with the address bar displaying simms-teach.com/cis90calendar.php. The page title is "Rich's Cabrillo College CIS Classes CIS 90 Calendar". The main content area is titled "CIS 90 (Fall 2014) Calendar" and includes a sidebar with a "CIS 90" link. The main table lists lessons with columns for Lesson, Date, Topics, and Link. The first lesson, Lesson 1, is highlighted and contains the following details:

Lesson	Date	Topics	Link
1	9/2	Class and Lesson Overview - Understand how this course will work - High-level overview of computers, operating systems, and virtual machines - Overview of UNIX/Linux market and architecture - Using SSH for remote network access - Using terminals and the command line Materials Presentation slides (download) Supplemental - PowerPoint: Logging into Opus (download) Assignments Student Survey Lab 1 CCC Confer Enter virtual classroom	5:15-6:00 PM 2-4 9/2-3 9/2-3 (9/2)

1. Browse to:
<http://simms-teach.com>
2. Click the **CIS 90** link.
3. Click the **Calendar** link.
4. Locate today's lesson.
5. Find the **Presentation slides** for the lesson and **download** for easier viewing.
6. Click the **Enter virtual classroom** link to join CCC Confer*
7. Log into Opus with Putty or ssh command.

* First time online students should use:
<http://www.cccconfer.org/support/Readiness>
to verify their computer is ready for CCC Confer.

Note: Blackboard Collaborate Launcher only needs to be installed once. It has already been downloaded and installed on the classroom PC's.

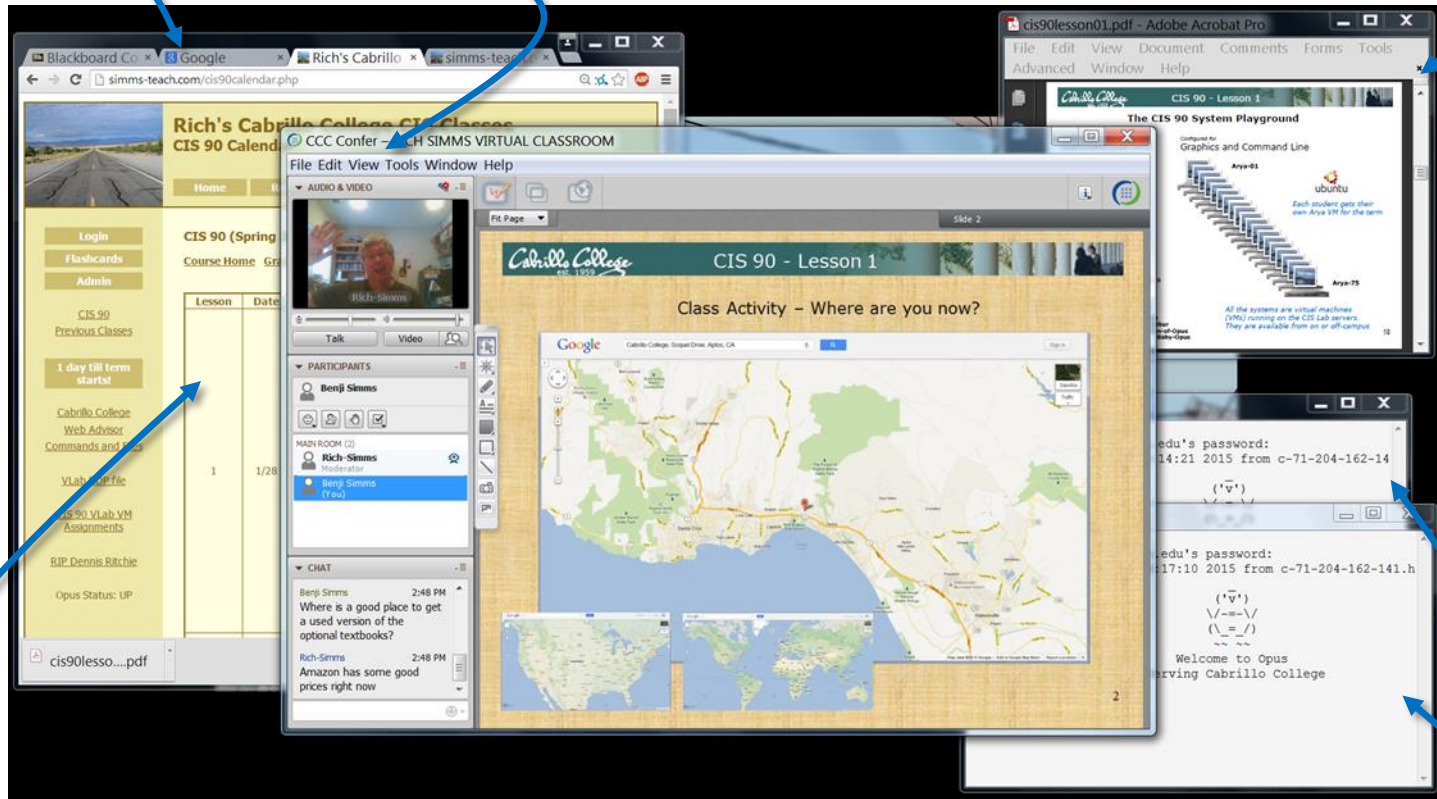


Student checklist for suggested screen layout

☐ Google

☐ CCC Confer

☐ Downloaded PDF of Lesson Slides



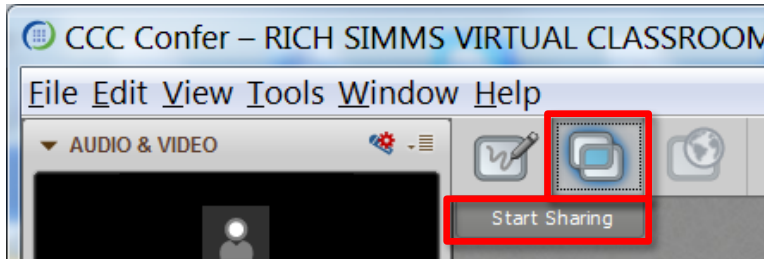
☐ CIS 90 website Calendar page

☐ One or more login sessions to Opus

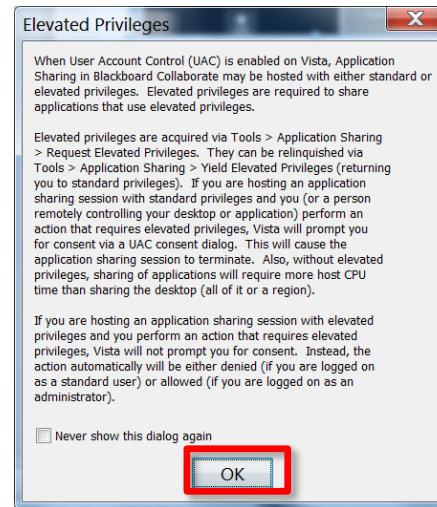


Student checklist for sharing desktop with classmates

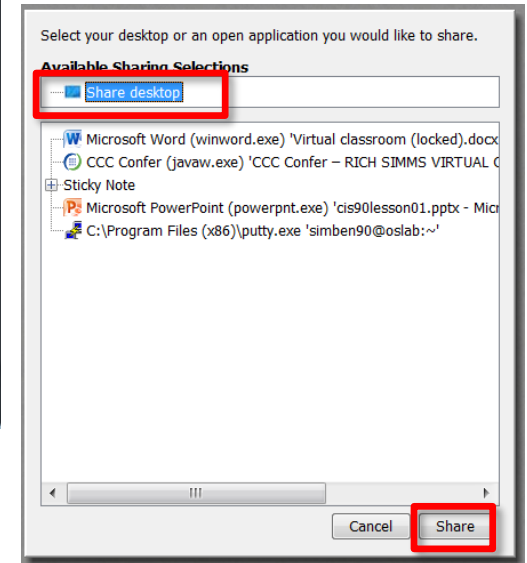
1) Instructor gives you sharing privileges



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



3) Click OK button.



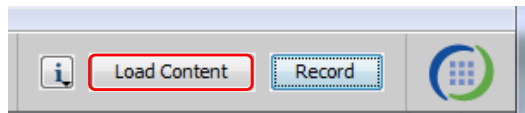
4) Select "Share desktop" and click Share button.



Rich's CCC Confer checklist - setup

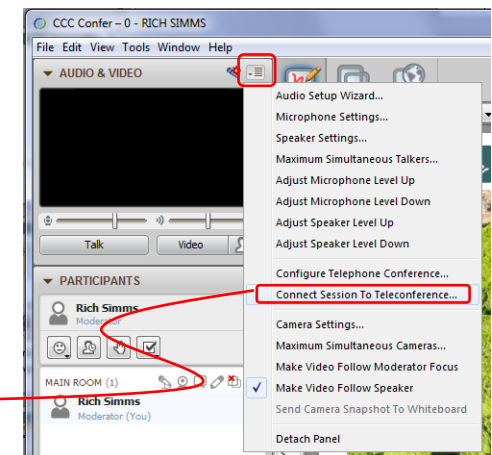
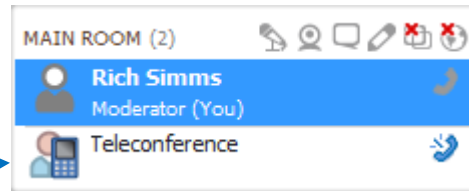


[] Preload White Board



[] Connect session to Teleconference

Session now connected to teleconference



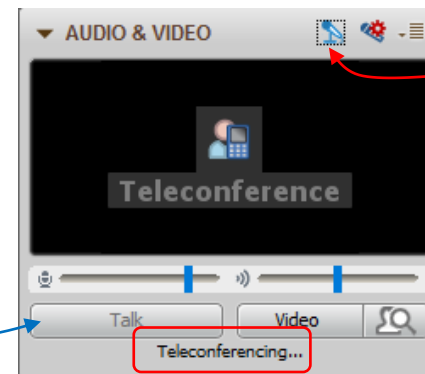
[] Is recording on?



Red dot means recording

[] Use teleconferencing, not mic

Should be grayed out



Should change from phone handset icon to little Microphone icon and the Teleconferencing ... message displayed



Rich's CCC Confer checklist - screen layout



The screenshot displays a Windows desktop environment during a CCC Confer session. The desktop includes several open applications:

- CCC Confer - 0 - RIC...:** A window showing the conferencing interface with a video feed of Rich Simms, a list of participants (Rich Simms, Moderator), and a chat window.
- simms-teach.com/docs/cis90/cis-90-TEST-1-Fall-12.pdf:** A web browser window displaying a quiz titled "Part 1 - Flashc (1 point each)". The quiz questions are:
 - [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]
- Putty:** A terminal window showing a login session for "simben90@oslab:". The output includes:


```
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 2011 from 10.10.10.10
```
- vSphere Client:** A window showing the vCenter interface, displaying a list of virtual machines under the "CIS 192" cluster.

Red callout boxes with arrows point to specific elements:

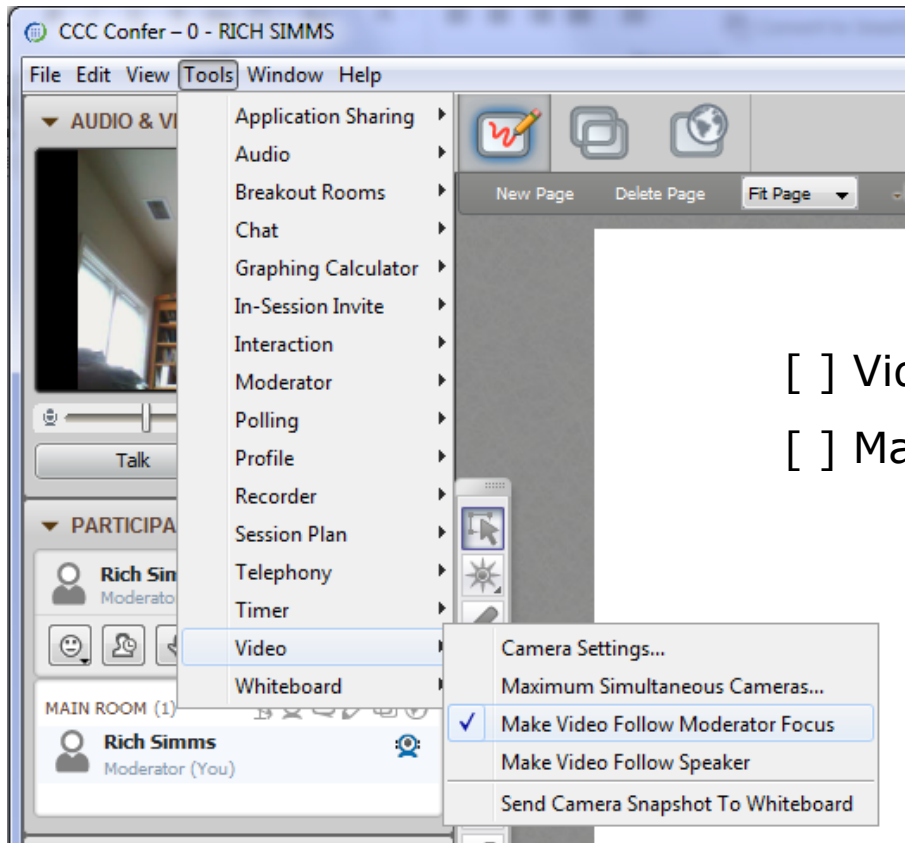
- foxit for slides:** Points to the "cis90lesson07.pdf" window in the background.
- chrome:** Points to the "simms-teach.com" browser window.
- putty:** Points to the terminal window.
- vSphere Client:** Points to the vCenter interface window.

[] layout and share apps





Rich's CCC Confer checklist - webcam setup

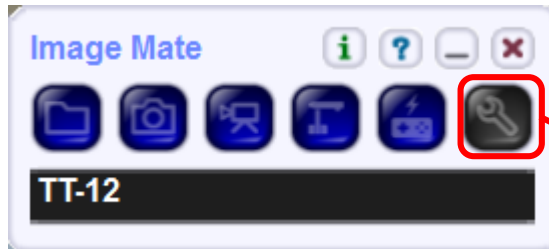


☐ Video (webcam)

☐ Make Video Follow Moderator Focus



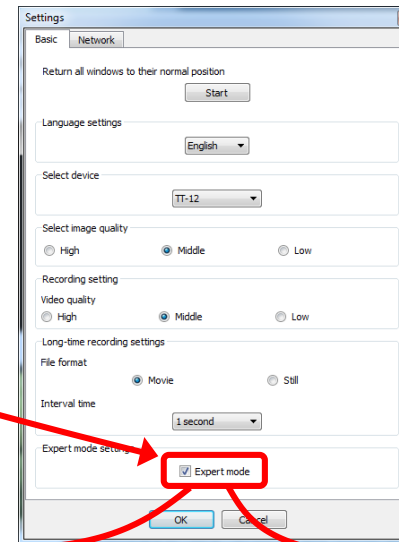
Rich's CCC Confer checklist - Elmo



Elmo rotated down to view side table



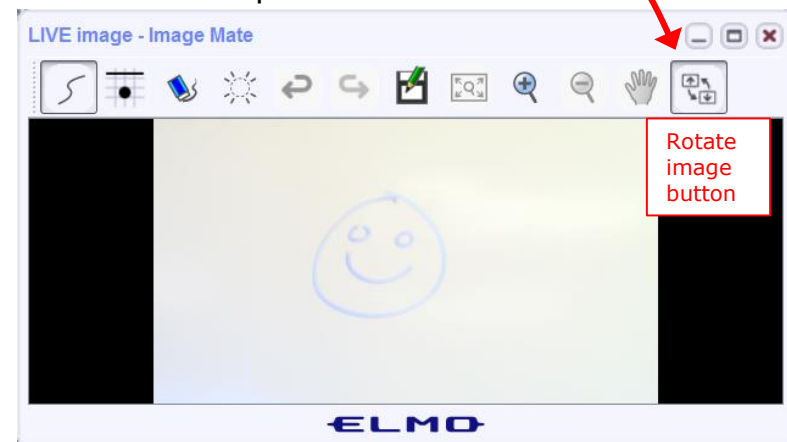
Run and share the Image Mate program just as you would any other app with CCC Confer



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

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

Elmo rotated up to view white board



Rich's CCC Confer checklist - universal fixes

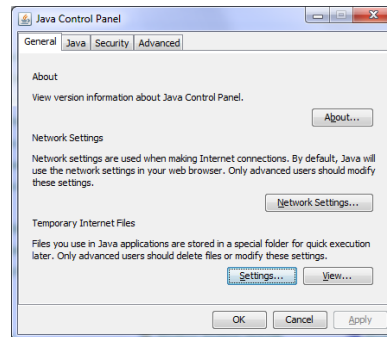
Universal Fix for CCC Confer:

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

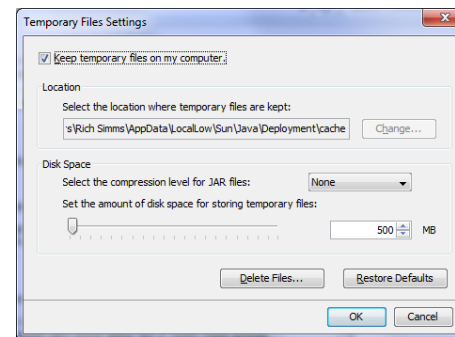
Control Panel (small icons)



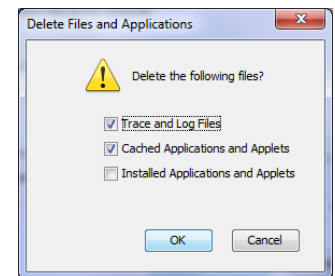
General Tab > Settings...



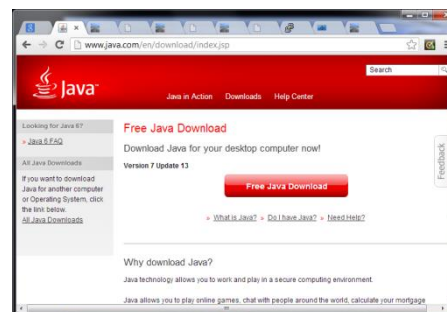
500MB cache size



Delete these

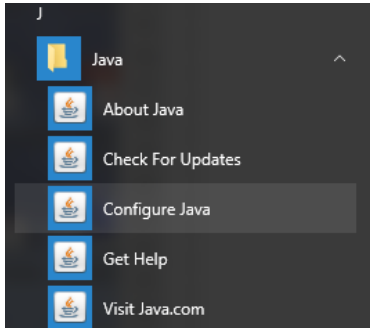


Google Java download

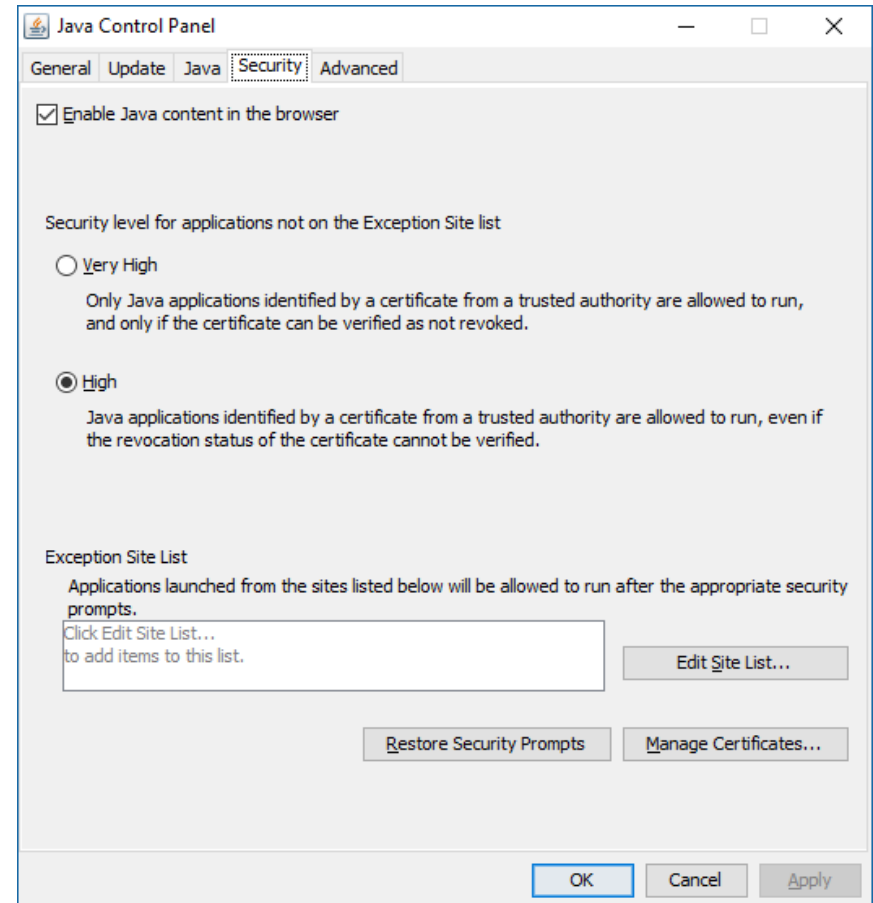




Rich's CCC Confer checklist - digital certificate work around



1. Open the [Java Control Panel](#)
2. Select the **Security** tab
3. Select **Edit Site List...**
4. Select **Add**
5. Click into the white box next to the red exclamation mark and type **<https://na-downloads.illuminate.com>**
6. Press OK
7. Press **Continue** on the pop-up message
8. Press OK
9. Access your session or recording once more



Start

Sound Check

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

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

Volume

**4 - increase conference volume.*

**7 - decrease conference volume.*

**5 - increase your voice volume.*

**8 - decrease your voice volume.*



Instructor: **Rich Simms**

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

Passcode: **136690**



Melissa



Tess



Alex



Daniel



Ian J.



Harold



Victor



Jasen



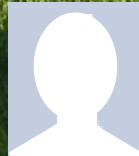
Sam



Steven P.



Ryan



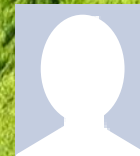
Roberto



Hans



Cristian



Nicholas



Fritz



Luis



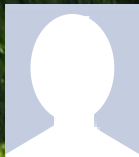
Nigel



Philip



Josh M.



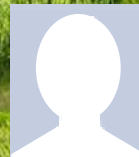
Ian K.



Gracie



James



Julian



Ken



Joshua V.



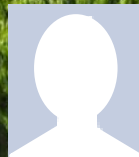
Samantha



Justin



Ian C.



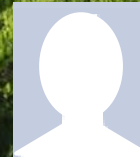
Venus



Stephen L.



Alison



Dillon

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 where account information is kept. • Understand why strong passwords are important. • Learn where commands are located. • Understand how the shell works to run commands. • Discover where to find documentation.	• Quiz • Questions • Using VLab • Virtual terminals • Logging in • Passwords • Housekeeping • Lesson 2 commands • The path • Location of common commands • Programs • Inputs to commands • Command syntax • Parsing • Variables • The shell (six steps) • Metacharacters • Shortcuts • Life without a path • Docs • Wrap up

Class Activity

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

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

If you haven't already,
log into Opus

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.

Extra Credit

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

For some flexibility, personal preferences or family emergencies there is an additional 90 points available of extra credit activities.

On the forum

Be sure to monitor the forum as I may post extra credit opportunities without any other notice!

On Lab 1 submittal

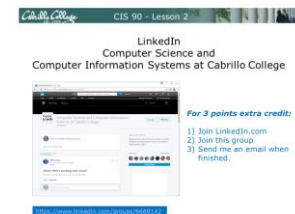
```
simben90@oslab:~
Lab 1 Scavenger Hunt
Update the table below with your collected items the

SYSTEM      ITEM      COLLECTED
defiant     star      <no entry>
lexington   instrument <no entry>
enterprise   movie     <no entry>
intrepid    fruit     <no entry>
freedom     book      <no entry>
excalibur   dog       Redbone Coonhound

BONUS QUESTION ANSWERS
Q1)
Q2)
Q3)

SELECTION MENU
1) Set star
2) Set instrument
3) Set movie
4) Set fruit
5) Set book
6) Set dog
7) Answer bonus questions
8) Submit your work for grading and quit
9) Quit without submitting
Enter selection (1-9):
```

In lesson slides



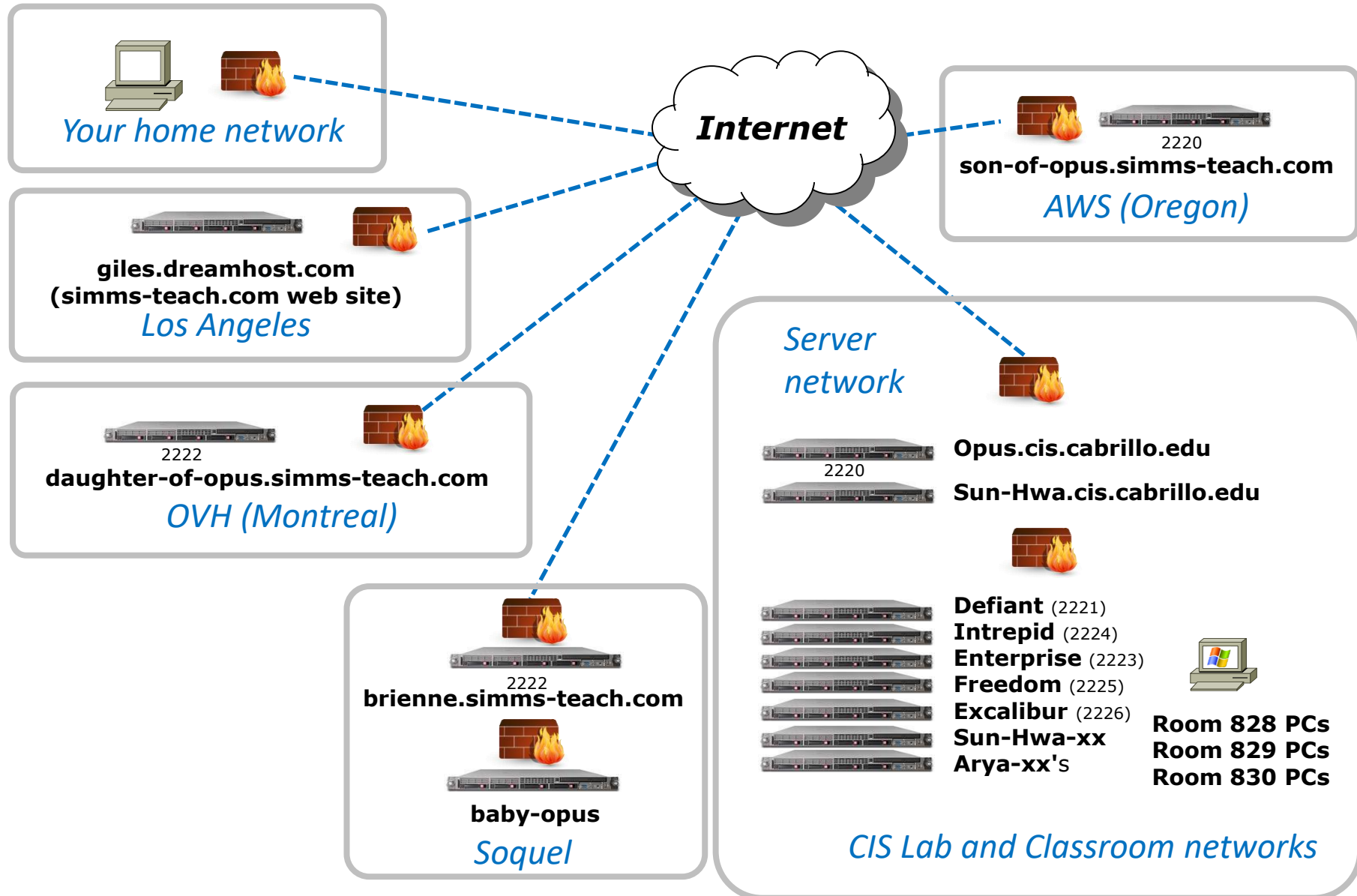
<http://simms-teach.com/cis90extracredit.php>

- **Web site content review** - The first person to email the instructor pointing out an error or typo on this website will get one point of extra credit for each unique error. The email must specify the specific document or web page, pinpoint the location of the error, and specify what the correction should be. Duplicate errors count as a single point. This does not apply to pre-published material than has been uploaded but not yet presented in class. (Up to 20 points total)



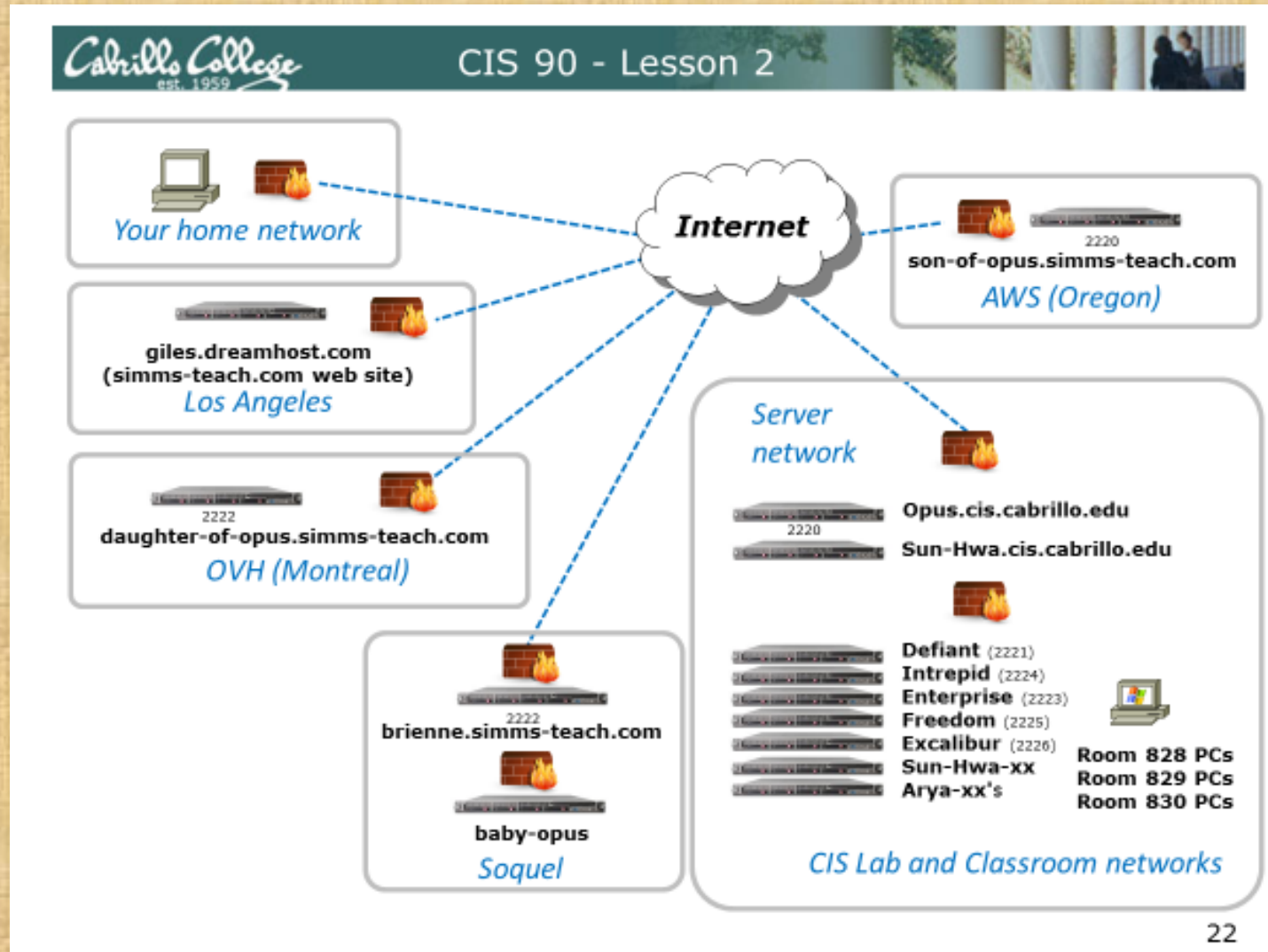
Navigating the Internet using SSH

Second driving lesson



Class Activity

Follow me if you can!



Using CIS VLab (Virtual Lab)

Third driving lesson

Command Line vs Graphical Desktop

Access the UNIX/Linux systems using:

ssh when:

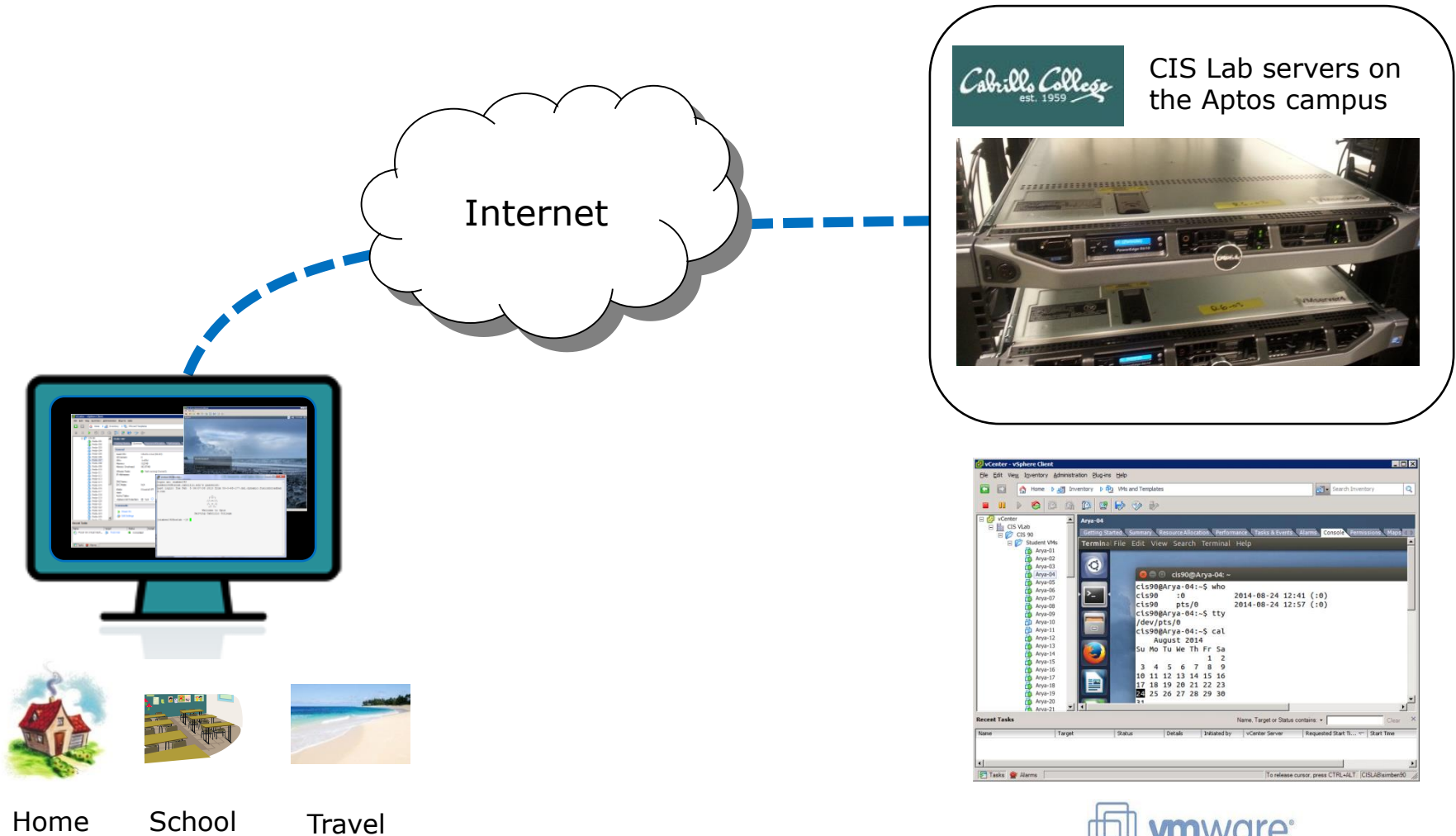
- You just need a command line
- Have a low or high speed network connection
- Note: Windows users can use Putty

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

Accessing CIS VLab VMs



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

Rich's Cabrillo College CIS Classes Home Page

Home Resources Forums CIS Lab Canvas

Login
Flashcards
Admin

CIS 76
CIS 90
Previous Terms

10 days till term starts!

Cabrillo College
Web Advisor
Blackboard
Commands and Files

VLab (classic)
VLab (web)
NETLAB+

CIS 76 VLab Pod Assignments

CIS 90 VLab VM Assignments

RIP Dennis Ritchie

Opus Status: UP

Rich Simms

Contact

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

My Fall 2016 Cabrillo Classes

- CIS 76 - Introduction to Information Assurance (Ethical Hacking) - [preview](#)
- CIS 90 Introduction to UNIX/Linux - [preview](#)

CIS 90 VLab Assignments

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

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

Accessing CIS VLab via vSphere Web Client

[Cabrillo College Web Advisor](#)
[Blackboard](#)
[Commands and Files](#)
[VLab \(classic\)](#)
[VLab \(web\)](#)
[NETLAB+](#)
[CIS 76 VLab Pod Assignments](#)

Contact
 • Email
 • Office

My Fall
 • CIS 76 - Introduction to Information Assurance (Ethical Hacking)
 • CIS 90 Introduction to UNIX/Linux - [preview](#)

vCenter Inventory Lists

Select Virtual Machines

Select your Arya VM

Summary tab, click on mini-console

Login

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

Your Arya Linux system is a VMware virtual machine. You will use VMware vCenter to access the VM's console.

From the console you can access the graphical desktop or the virtual terminals.

You may have to power-on your VM if it has been shutdown. One way is to right-click on your VM's name in the list and select Power > Power On.

Accessing CIS VLab via vSphere Client

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

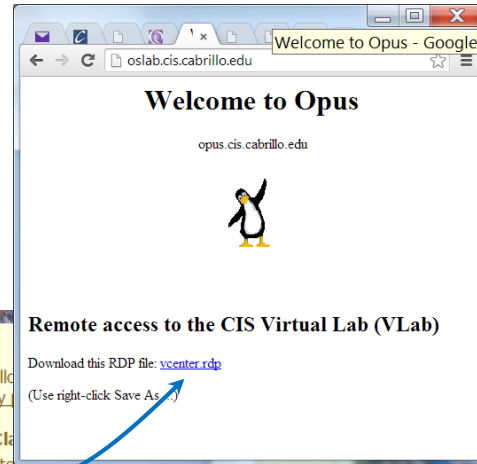
Cabrillo College
Web Advisor
Blackboard
Commands and Files
VLab (classic)
VLab (web)
NETLAB+
CIS 76 VLab Pod
Assignments

Contact

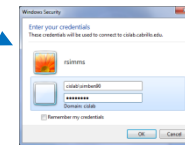
- Email: risimms@cabrillo.edu
- Office hours: [directory](#)

My Fall 2016 Cabrillo Cl

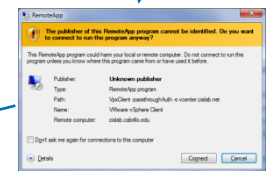
- CIS 76 - Introduction to Information Assurance (Ethical Hacking)
- CIS 90 Introduction to Unix/Linux - [preview](#)



Open



Login



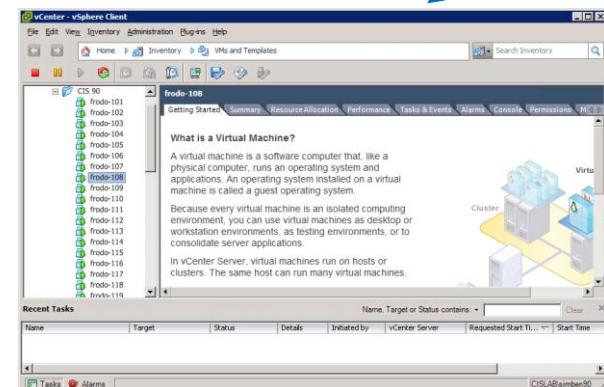
Connect



Ignore



Wait ...



Locate and select your assigned VM

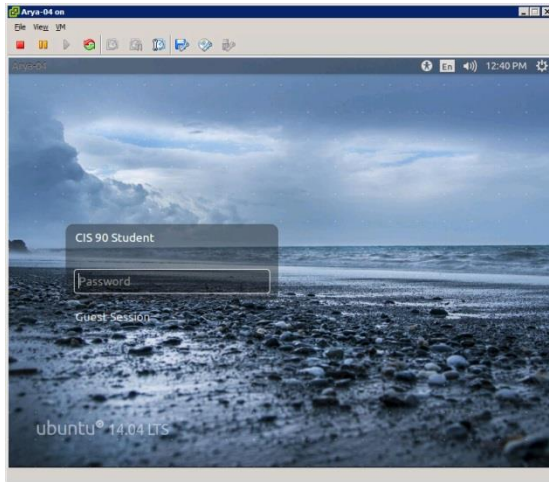
- 1) Download the vcenter.rdp file to your desktop and then open it to access VLab.
- 2) Mac users need to install an RDP app like Microsoft Remote Desktop.
- 3) When entering your username and password you must preface your username with the "cislab\", for example Benji would use: cislab\simben90

Class Activity

Follow the instructor to open a console on your VM

- ☐ Browse to <http://simms-teach.com>
- ☐ Determine which Arya VM is yours
- ☐ Connect to VLab's vCenter using the Web Client
- ☐ Navigate to CIS 90 Arya VMs
- ☐ Select your VM and open the console

Log in as
CIS 90 Student

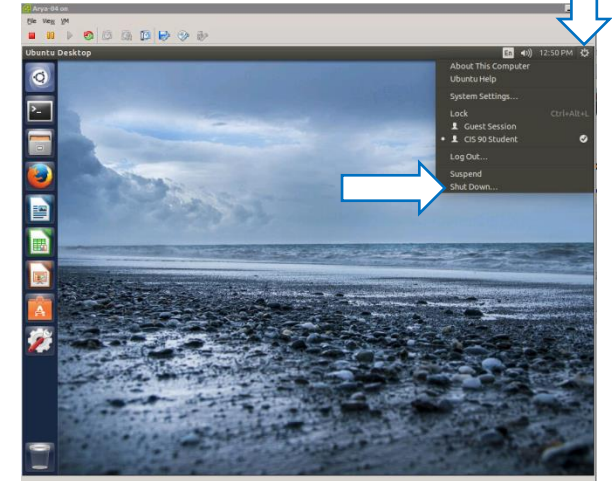


The Arya VM

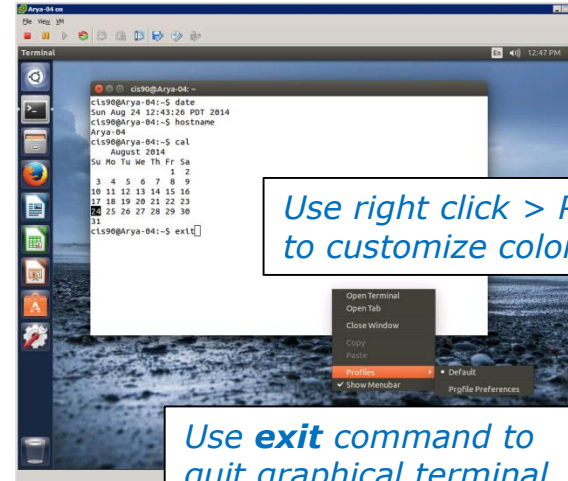
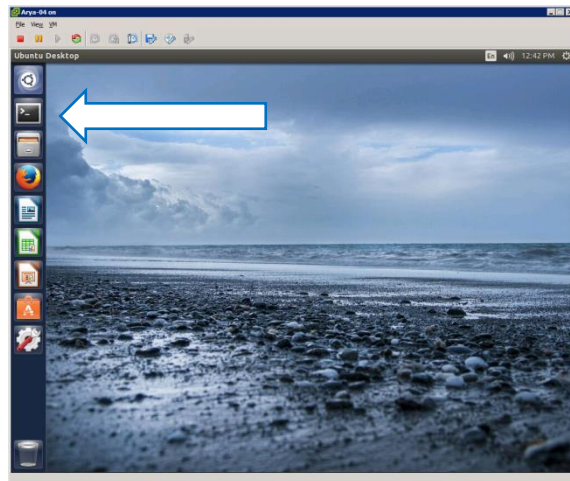


ubuntu

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



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



*Use right click > Profiles
to customize colors*

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

Class Activity

Follow the instructor to login and use your VM

- ☐ Login to your Arya VM*
- ☐ Open a graphical terminal
- ☐ Use who command to see logins
- ☐ Find the "toothed gear" icon to logoff, restart or shutdown

*See the CIS 90 welcome email or announcement in Canvas from the instructor for Arya login credentials



Virtual Terminals (consoles)

Fourth driving lesson

Virtual Terminals

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

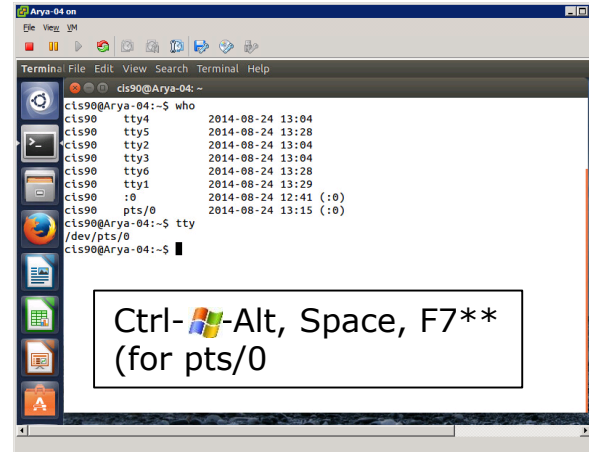
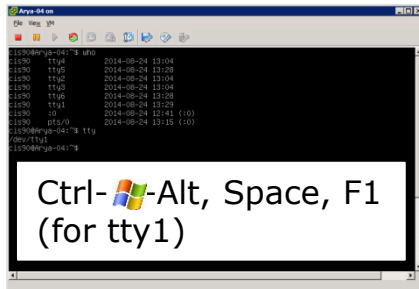
-Alt-Space-F2 (for tty2)'." data-bbox="29 125 375 353"/>

-Alt-Space-F3 (for tty3)'." data-bbox="29 353 428 579"/>

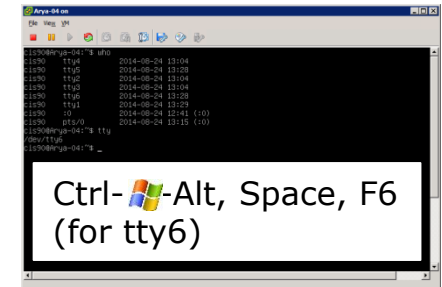
-Alt-Space-F4 (for tty4)'." data-bbox="29 579 491 977"/>

-Alt-Space-F7 (for pts/0)'." data-bbox="508 520 922 977"/>

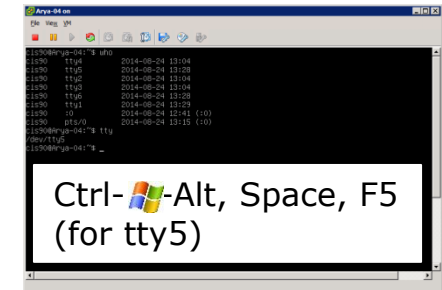
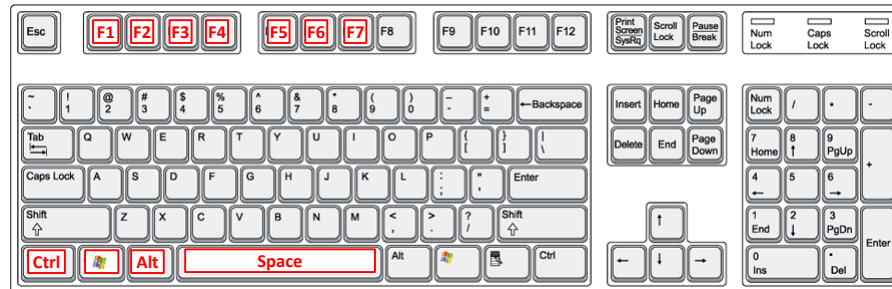
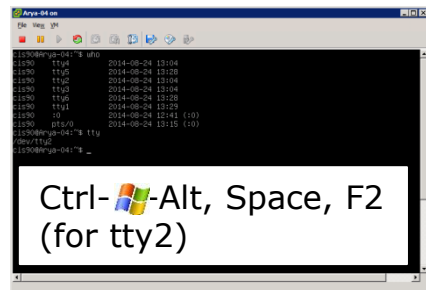
Changing Virtual TTY Terminals using **VMware vSphere**



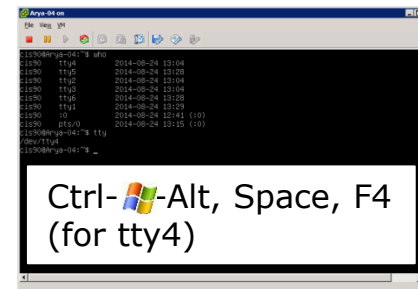
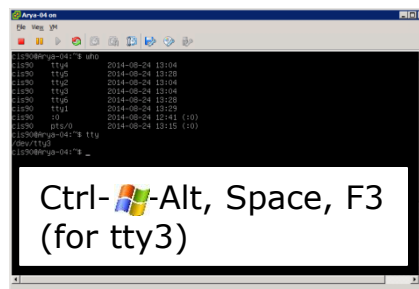
Windows PC Keyboard



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



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



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



Changing Virtual Terminals on VMware Linux VMs

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

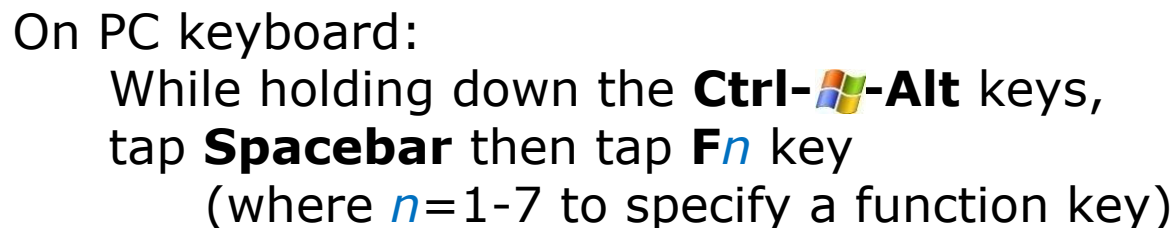
Pressing the  on some Windows keyboards may not be necessary

F7 is graphics mode for the Ubuntu VMs.

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

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

Changing Virtual Terminals with a PC keyboard



VMware VM Operations

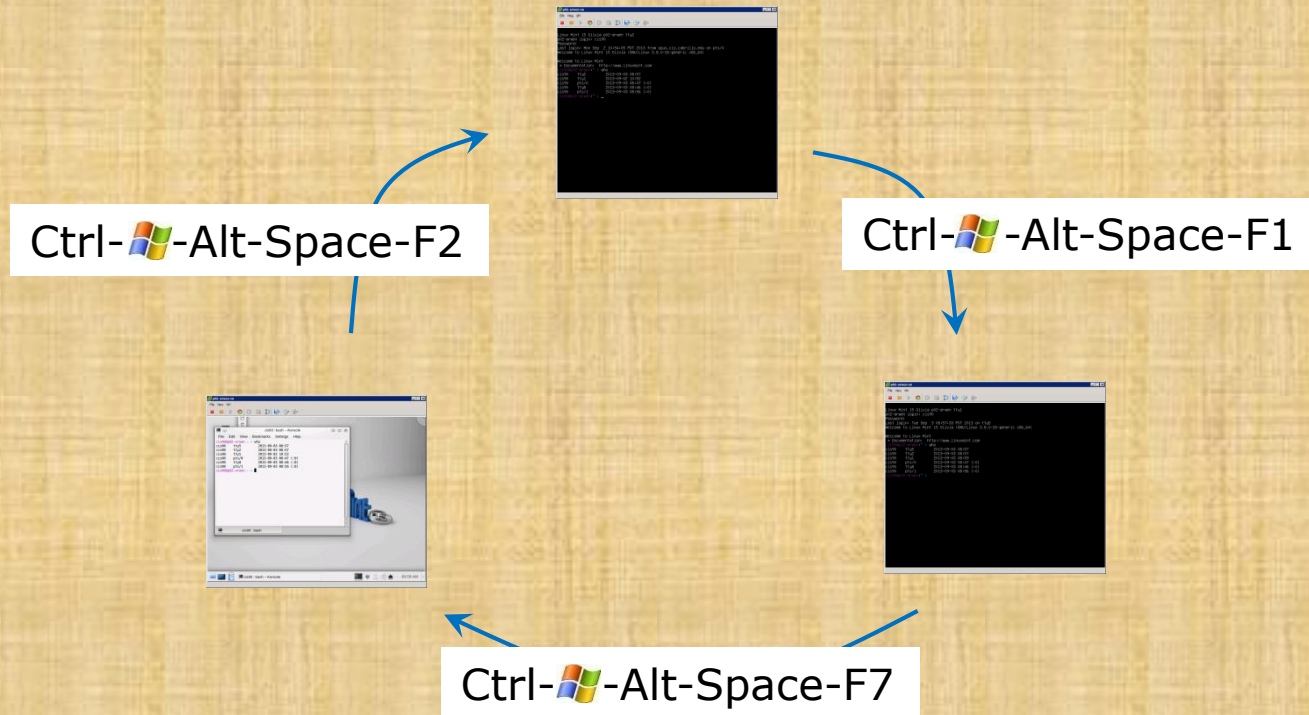
Changing Virtual Terminals with a Mac keyboard



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)

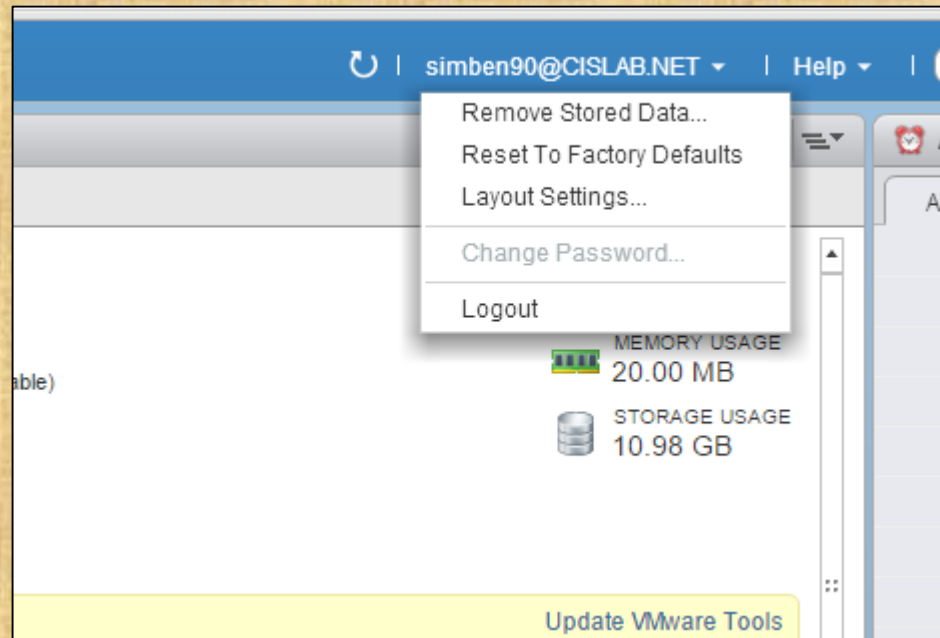
Class Activity



Follow the instructor to:

- 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

Class Activity



Logout of Vlab's vCenter

Your VM will keep running even though you disconnect from vCenter

Logging In (authentication)



Who goes there?

What's the password?

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

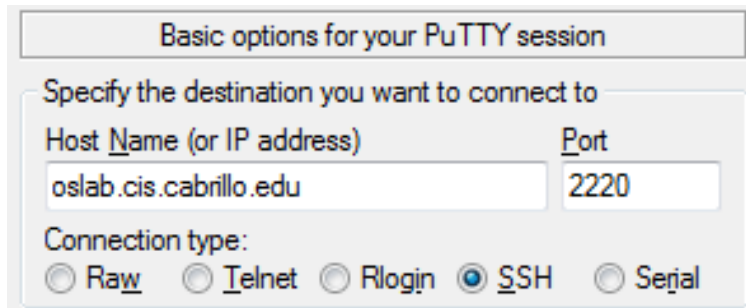
Logging in

- A system administrator can create user accounts for each user that is allowed to login
- To login you must be authenticated as one of those users
- There are two common authentication methods used:
 - 1) Username and password
 - 2) Public & private keys

We will cover just usernames and passwords today

Logging in

Logging in using Putty from Windows PCs



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address) Port

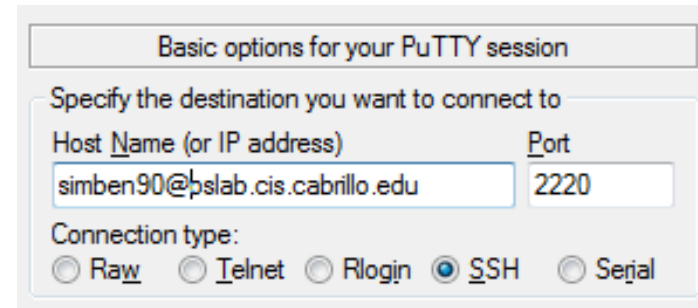
oslab.cis.cabrillo.edu 2220

Connection type:

☐ Raw ☐ Telnet ☐ Rlogin ☒ SSH ☐ Serial

If you don't specify your username the system will prompt you for both your username and password

```
login as: simben90
simben90@oslab.cis.cabrillo.edu's password:
```



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address) Port

simben90@oslab.cis.cabrillo.edu 2220

Connection type:

☐ Raw ☐ Telnet ☐ Rlogin ☒ SSH ☐ Serial

If you specify your username the system will just prompt you for your password

```
Using username "simben90".
simben90@oslab.cis.cabrillo.edu's password:
```

Logging in with the ssh command from Mac or UNIX/Linux systems

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

If you don't specify a username the ssh command will use your current username. Be careful, that username may not exist on the remote system you are trying to login to.

```
[rsimms@daughter-of-opus ~]$ ssh -p 2220 simben90@oslab.cis.cabrillo.edu
simben90@oslab.cis.cabrillo.edu's password:
```

Logging in

Logging in on a virtual terminal

```
CentOS release 6.5 (Final)
Kernel 2.6.32-504.16.2.el6.i686 on tty1

oslab login: simben90
Password:
Last login: Tue Sep  8 16:02:07 from 2607:f380:80f:f830:250:56ff:febd:3193

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

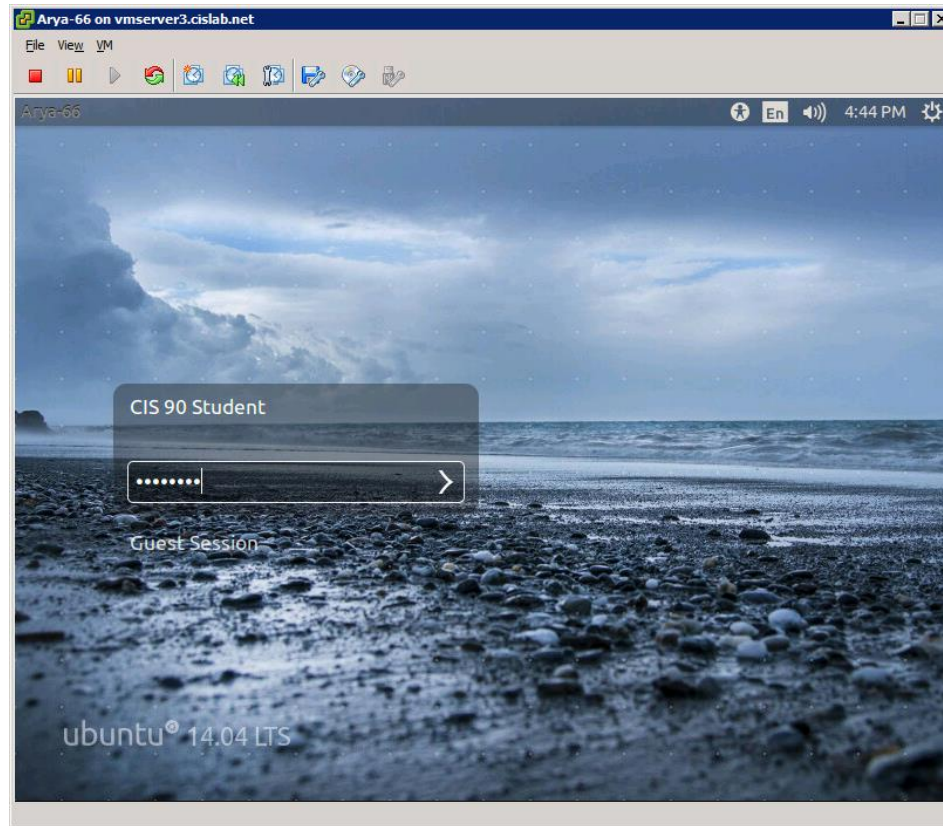
Welcome to Opus
Serving Cabrillo College

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

When you have direct physical access to a system you can use one of these virtual terminals on the system console. You are not using ssh over the network in this situation.

Logging in

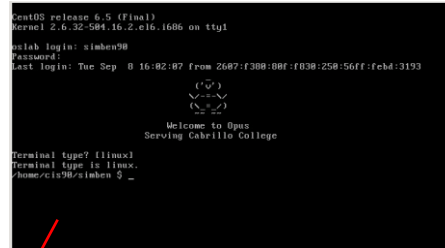
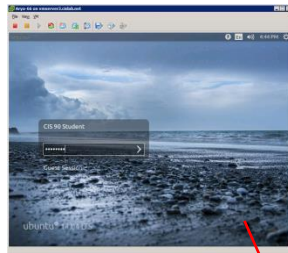
Logging in using a graphical desktop (Ubuntu)



This can be done locally or over the network

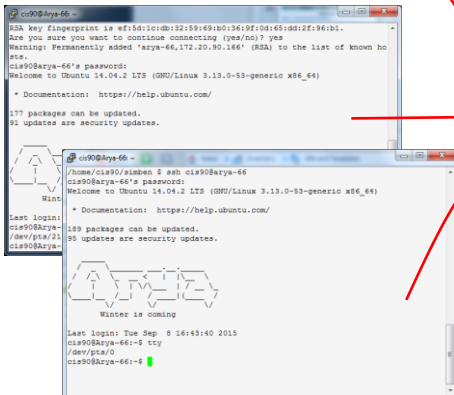
Just for kicks

:0



tty1

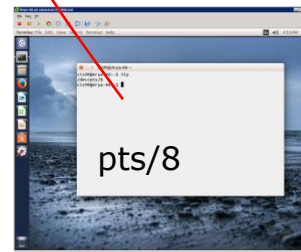
pts/21



pts/0

cis90@Arya-66:~\$ who

cis90	tty1	2015-09-08 16:43
cis90	:0	2015-09-08 16:53 (:0)
cis90	pts/21	2015-09-08 16:39 (opus.cis.cabrillo.edu)
cis90	pts/0	2015-09-08 16:55 (opus.cis.cabrillo.edu)
cis90	pts/8	2015-09-08 16:53 (:0)



pts/8

Let's login to an Arya using a virtual terminal, a graphical desktop, two ssh sessions and a graphical terminal on the graphical desktop

Logging in

- For systems that are not connected to a directory service (e.g. Microsoft Active Directory) all user accounts are kept in a file named **/etc/passwd**
- For systems that are not connected to a directory service all passwords are kept encrypted in a file named **/etc/shadow**

The `/etc/passwd` file

The SUPER user is named root

```
[rsimms@daughter-of-opus ~]$ cat /etc/passwd  
root:x:0:0:root:/root:/bin/bash
```

Snipped

```
deanna:x:2009:1701:Deanna Troi:/home/deanna:/bin/bash  
chakotay:x:2010:1701:Chakotay:/home/chakotay:/bin/bash  
kira:x:2011:1701:Kira Nerys:/home/kira:/bin/bash  
chekov:x:2012:1701:Pavel Chekov:/home/chekov:/bin/bash  
[rsimms@daughter-of-opus ~]$
```

To login your username must match one of the accounts in the `/etc/passwd` file

Note: this file no longer contains the passwords!

Viewing your account in `/etc/passwd`

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

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

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



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!*

The /etc/shadow file

The SUPER user is named root

```
[rsimms@daughter-of-opus ~]$ cat /etc/shadow
```

```
cat: /etc/shadow: Permission denied
```

```
[rsimms@daughter-of-opus ~]$ sudo cat /etc/shadow
```

*Use sudo to run command
as superuser (root)*

```
[sudo] password for rsimms:
```

```
root:$6$  
:16226:0:99999:7:::
```

Snipped

```
deanna:$6$hsAXq0Jk$ndIt.oxiFL/qZ7pLAFOaGgxpxAHDEj7ukpd0PfeRN0J9q07Z6Cg0V  
3hzo9eSAk0GlaywDtqwL5NefNEEwf9FR1:16686:0:99999:7:::
```

```
chakotay:$6$c/kFViIa$nTUJcvJRCut8PwvOSYLlopAI25UsFLNKerGF8OhQIkI78RHTXE1  
KOOwvDRSW6BAi4pui7LLpi6JP8QCBMVU1s1:16686:0:99999:7:::
```

```
kira:$6$3dqjzQCw$G2bJapsW07IhLD.cQfI9htk.hWiGUdJhOjNDxZT4zTN9lWTP0KDJ6eg  
hBzvT86xUXhIM8XDFB4WpOt.5Ab0jJ.:16686:0:99999:7:::
```

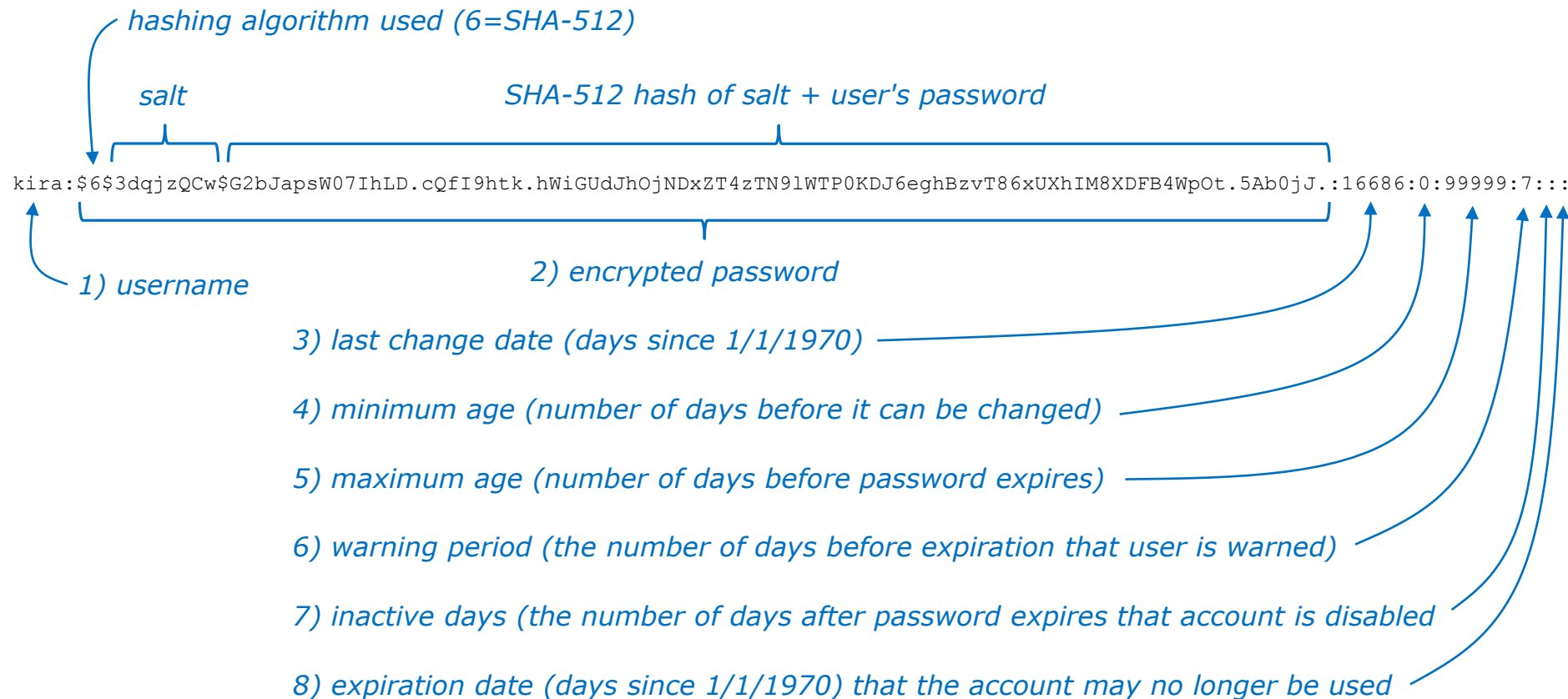
```
chekov:$6$jd4PMdv0$HPyW/k04DjMDeLO3qUfEzvQj0fWpLuUWMh9Rv1Ov1V3N/zQxhdhS3  
YfSLdhHz0rKBelwzGGx07CrzOfL3MKNa1:16686:0:99999:7:::
```

```
[rsimms@daughter-of-opus ~]$
```

To login, your password must match the encrypted account password kept in the /etc/shadow file

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

The /etc/shadow file



Note the major fields in /etc/shadow are delimited with a ":". The encrypted password field is further delimited with a "\$"

Class Activity

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

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

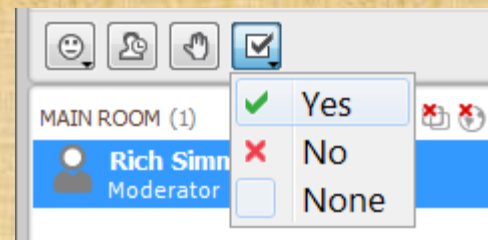
Note the field separator used in /etc/passwd is a ":"

1) Find your record in /etc/passwd

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

2) cat /etc/shadow

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



For Supplemental Study

<http://www.slashroot.in/how-are-passwords-stored-linux-understanding-hashing-shadow-utils>

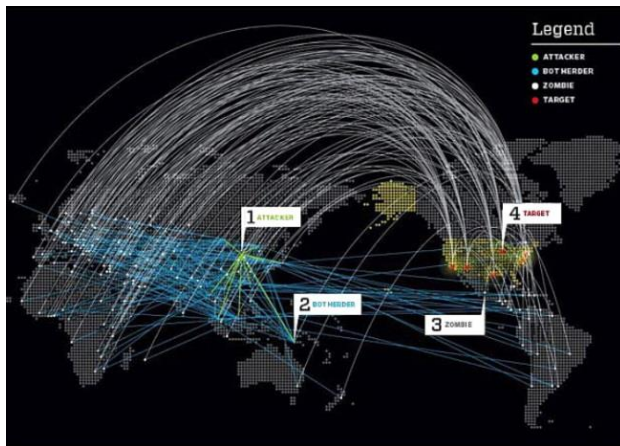


Excellent article on how passwords created and stored

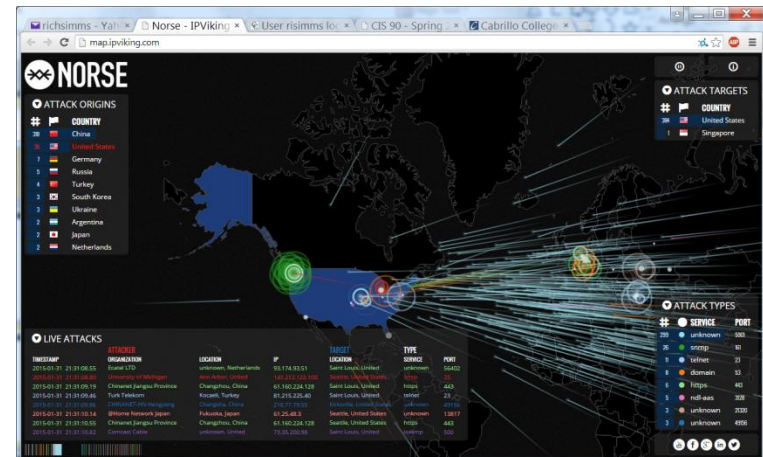
Passwords

Your password

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



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



<http://map.norsecorp.com/#/>

July 3, 2015 – Datacenter is idle over the summer but we still have lots of international visitors!

Top source countries

PA-500 : Friday, July 03, 2015

Source Country	Bytes	Sessions
172.16.0.0-172.31.255.255	2.84 G	79.68 k
192.168.0.0-192.168.255.255	7.54 M	36.23 k
Unknown	62.17 M	6.13 k
United States	209.74 M	4.20 k
China	26.66 M	1.13 k
Hong Kong	13.88 M	1.05 k
Russian Federation	92.51 M	884
France	62.30 M	827
Germany	16.16 M	460
Austria	875.47 k	404
United Kingdom	13.38 M	148
Ukraine	12.88 M	144
Spain	4.72 M	57
European Union	797.22 k	42
Israel	828.43 k	38
Korea Republic Of	1.85 M	33
Netherlands	321.20 k	31
Morocco	287.62 k	30
Switzerland	1.74 M	28
Thailand	14.03 k	24
Taiwan ROC	59.15 k	21
Virgin Islands British	1.54 M	21
Romania	281.80 k	17
Canada	393.89 k	16
Estonia	334.02 k	15

Tool: Palo Alto Networks PA-500 (one page of a daily report)

May 28, 2015 – Bad 3-way handshakes being sent to Opus from France

188.165.15.181 » Check and report abuse IP

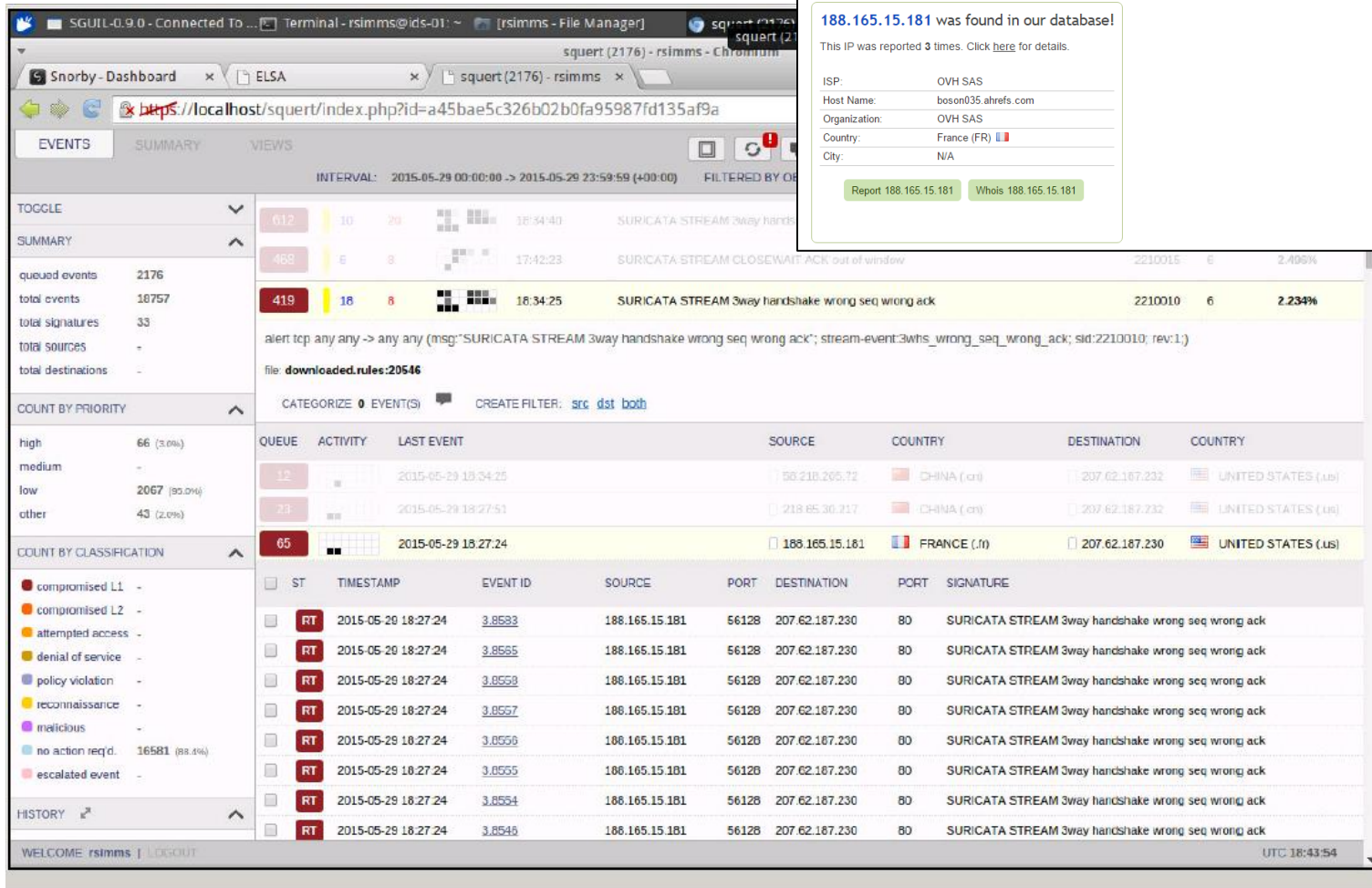
Enter an IP address or a Domain name:

Example: 207.46.197.32 or microsoft.com

188.165.15.181 was found in our database!

This IP was reported 3 times. [Click here](#) for details.

ISP: OVH SAS
Host Name: boson035.ahrefs.com
Organization: OVH SAS
Country: France (FR) 
City: N/A



The screenshot shows the Squert dashboard with the following sections:

- EVENTS**: A list of events with columns for ID, Count, and Description. The selected event is 419, with a count of 18, and the description "SURICATA STREAM 3way handshake wrong seq wrong ack".
- SUMMARY**: A summary of the event, including the alert message and the file path.
- COUNT BY PRIORITY**: A table showing the count of events by priority level.
- COUNT BY CLASSIFICATION**: A table showing the count of events by classification.
- HISTORY**: A table showing the history of events.

QUEUE	ACTIVITY	LAST EVENT	SOURCE	COUNTRY	DESTINATION	COUNTRY
12		2015-05-29 18:34:25	50.218.205.72	CHINA (cn)	207.62.187.232	UNITED STATES (us)
23		2015-05-29 18:27:51	218.65.30.217	CHINA (cn)	207.62.187.232	UNITED STATES (us)
65		2015-05-29 18:27:24	188.165.15.181	FRANCE (fr)	207.62.187.230	UNITED STATES (us)

ST	TIMESTAMP	EVENT ID	SOURCE	PORT	DESTINATION	PORT	SIGNATURE
RT	2015-05-29 18:27:24	3.8583	188.165.15.181	56128	207.62.187.230	80	SURICATA STREAM 3way handshake wrong seq wrong ack
RT	2015-05-29 18:27:24	3.8585	188.165.15.181	56128	207.62.187.230	80	SURICATA STREAM 3way handshake wrong seq wrong ack
RT	2015-05-29 18:27:24	3.8559	188.165.15.181	56128	207.62.187.230	80	SURICATA STREAM 3way handshake wrong seq wrong ack
RT	2015-05-29 18:27:24	3.8557	188.165.15.181	56128	207.62.187.230	80	SURICATA STREAM 3way handshake wrong seq wrong ack
RT	2015-05-29 18:27:24	3.8556	188.165.15.181	56128	207.62.187.230	80	SURICATA STREAM 3way handshake wrong seq wrong ack
RT	2015-05-29 18:27:24	3.8555	188.165.15.181	56128	207.62.187.230	80	SURICATA STREAM 3way handshake wrong seq wrong ack
RT	2015-05-29 18:27:24	3.8554	188.165.15.181	56128	207.62.187.230	80	SURICATA STREAM 3way handshake wrong seq wrong ack
RT	2015-05-29 18:27:24	3.8546	188.165.15.181	56128	207.62.187.230	80	SURICATA STREAM 3way handshake wrong seq wrong ack

July 9, 2015 – Datacenter is idle over the summer break but we still have lots of strangers trying to log in!

Threat Types

Top 5 Spyware

Spyware	Count
Morto RDP Request Traffic	13

Top 5 Vulnerabilities

Vulnerability	Count
LDAP: User Login Brute-force Attempt	12,302
MS-RDP Brute-force Attempt	3,369
SSH User Authentication Brute-force Atte..	9
PHP CGI Query String Parameter Handli...	6
PHP CGI Query String Parameter Handli...	6

Top 5 Viruses

No matching data found

Threat

Top 5 Attackers

Address	Count
cisvdc.cis.cabrillo.edu	12,302
162.242.228.100	3,186
195-154-157-104.rev.poneytelecom.eu	133
mail.vadimedical.com.tw	28
hosted-by.invisionarg.com	17

Top 5 Victims

Address	Count
rdserver.cis.cabrillo.edu	15,684
ed.cis.cabrillo.edu	11
opus.cis.cabrillo.edu	2
vcenter.cis.cabrillo.edu	2
pengo.cis.cabrillo.edu	2

Top 5 Attacker Countries

Country	Count
172.16.0.0-172.31.255.255	12,302
United States	3,210
France	133
Taiwan ROC	28
Netherlands	17

Tool: Palo Alto Networks PA-500 (one page of a daily report)

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
jimj:
  70.132.20.25 (adsl-70-132-20-25.dsl.anfc21.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.anfc21.sbcglobal.net): 1 time
rsimms:
  63.249.86.11 (dsl-63-249-86-11.cruzio.com): 2 times
```

Tool: logwatch report showing malicious attempts to break into Opus

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

```
[root@opus log]# lastb | sort | cut -f1 -d' ' | grep -v ^$ | uniq -c > bad
```

```
[root@opus log]# sort -g bad > bad.sort
```

```
[root@opus log]# cat bad.sort | tail -50
```

471	ftp	
472	public	
490	test	
490	tomcat	610 test
498	user	656 noc
506	service	686 www
508	mike	690 postfix
508	username	723 john
524	cyrus	734 testing
530	pgsql	738 adam
532	test1	746 alex
544	master	754 info
554	linux	798 tester
554	toor	832 library
576	paul	935 guest
584	support	990 admin
590	testuser	1002 office
604	irc	1022 temp
		1070 ftpuser
		1138 webadmin
		1298 nagios
		1332 web
		1374 a
		1384 student
		1416 postgres
		1690 user
		1858 oracle
		1944 mysql
		2086 webmaste
		5324 test
		10803 root
		10824 admin
		18679 root
		24064 root

```
[root@opus log]#
```

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

How to make a strong password

Current goal: require at least 2^{64} guesses

- Use upper case, lower case, punctuation, digits
- The longer the better (10 or more characters) $94^{10} \Rightarrow 65.64$ bits of entropy
- Random, not in any dictionary
- Something you can remember (Google "best password managers")
- Different password for different services
- Keep it secret -- change when compromised
- A MUST for your email accounts!

GOOD (but not truly random)

Wh0le#!!!!	(Whole sh'bang)
KuKu4 (co) 2	(Cuckoo for Cocoa Puffs)
#0p.&.s@ve	(shop and save)
Idl02\$d@y	(I do laundry on Tuesday)
Iwb@tB0aWw	(<u>I</u> <u>w</u> as <u>b</u> orn <u>a</u> t the <u>b</u> ottom of a <u>w</u> ishing <u>w</u> ell)

BETTER (pass phrases of 6 random words) $2000^6 \Rightarrow 65.79$ bits of entropy

splendid roll arrest boiling silk shelter
heap pancake wooden complete inject ethereal
few balance note sedate alike tense

passwd command

Change user's password

Syntax:

passwd [username]

Example:

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

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

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

John the Ripper

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



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

Four users: deanna, chakotay, kira and chekov with weak passwords:

1234567
secret
terces
chekov1

```
[sudo] password for rsimms:
deanna:$6$M9MSUz0p$wfnU/Hbv86hG/Sbi0v9aaCl.bXhQixQd7qG0wrpGsAJUzU5Bum2QiBz9uTf7m
/IgwaZdImImuMIe7UX/yfFru.:2009:1701:Deanna Troi:/home/deanna:/bin/bash
chakotay:$6$eDZrKrit$gHcZ6zJnywZ5.XGSE60s53q4UJQoGDdEmjEk7k6R1hVZNv7zWtle9tXhWvE
NkfqZft2bmCNGaKwvAUN4MM2.v.:2010:1701:Chakotay:/home/chakotay:/bin/bash
kira:$6$1KD.GMs6$PJMd77APM05u6fFdFTpxoU2CEMLyQl1hDUQkC64kfxjgx/hXgU0Q5o/Lxuh80
Ob0g6tYbsXkR6fQAi5R0JF0:2011:1701:Kira Nerys:/home/kira:/bin/bash
chekov:$6$fj9vDNMO$JH9vCmNIIfKY1kTlw/L05ynBHaeLrBV5i49cIcrnnT2W7ioCncWtX07pvnZ1pb
vu1Yp8ziSrEKsp3RoqLzXEbm.:2012:1701:Pavel Chekov:/home/chekov:/bin/bash
[rsimms@sister-of-opus ~] $ john-run
Start cracking passwords? (press Enter to continue)
```

```
Wed Sep  7 10:21:58 PDT 2016
```

```
Warning: detected hash type "sha512crypt", but the string is also recognized as
"crypto"
Use the "--format=crypt" option to force loading these as that type instead
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA5
12 64/64 OpenSSL])
Warning: OpenMP is disabled; a non-OpenMP build may be faster
Press 'q' or Ctrl-C to abort, almost any other key for status
chekov1      (chekov)
secret       (chakotay)
1234567      (deanna)
-
```

sister-of-opus

For Supplemental Study

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

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

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

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

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

The Password Haystack Concept in 150 Seconds
Los Angeles' KABC-TV produced a terrific 15-second and a half-minute explanation of the Password Haystacks concept. Click this link to view their quick introduction.

GRC's Interactive Brute Force Password "Search Space" Calculator
(WARNING: you do have your needle your browser. What happens then, stays there.)

No Uppercase No Lowercase No Digits No Symbols 5 Characters

dummy

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

Brute Force Search Space Analysis:

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

Time Required to Exhaustively Search this Password's Space:

Online Attack Scenario: (Assuming one thousand guesses per second)	3.43 hours
Offline Fast Attack Scenario: (Assuming one hundred billion guesses per second)	0.000124 seconds
Massive Cracking Array Scenario: (Assuming one hundred trillion guesses per second)	0.00000124 seconds

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

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

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

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

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

Password strength calculator for random passwords

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

Why Passwords Fail

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

CMPS 485: Password Complexity

Ryan Riley

Subscribe 24

88 views

Excellent presentation on making strong passwords

Best Practices

Beginners guide to beefing up your online privacy and security



<http://arstechnica.com/security/2016/12/a-beginners-guide-to-beefing-up-your-privacy-and-security-online/>

- Install updates (especially browser and OS).
- Use strong passwords and passcodes.
- Encrypt your phones and computers.
- Use two-factor authentication.
- Use a password managers (example products: 1Password and LastPass).
- Encrypt SMS and voice calls (example products, Signal).
- Use VPNs on public Wi-Fi (example services, Private Internet Access).
- Secure end-to-end email (example ProtonMail).
- Delete old emails.
- For more in-depth strategies see EFF's Surveillance Self-Defense page.

<https://ssd.eff.org/>

Housekeeping



Housekeeping

1. Your student survey is due today
2. Lab 1 due by 11:59PM (Opus time) tonight

Use **submit** to turn in your work

Grading Rubric (30 points)

5 points for each correct scavenger hunt item

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

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

3. Last day to drop/add is this Saturday

Add Codes

- Available after class (stop by or email me).
- Please use them online ASAP!
- If you missed the first and second class obtaining an Add code will be conditional on catching up by the College's last day to add:
 - a) making a forum post.
 - b) answering one of the "first minute" Quiz 2 questions.
 - c) submitting the survey (part of the first assignment).
 - d) completing Lab 1 (no credit if turned in after due date).
 - e) completing at least Steps 1-2 in Lab 2.

Roll Call

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

Turn off recording

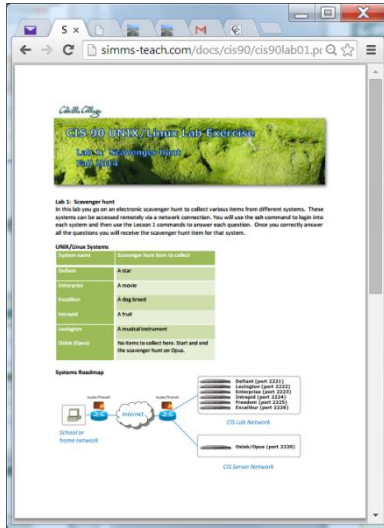
Do roll call using
both rosters

Turn on recording

Lab Assignments

Pearls of Wisdom:

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



Grading Code Names Lord of the Rings Characters

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

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

Introduction to UNIX/Linux (CIS 90)
Student Survey

Student Information

- Preferred first name: _____ Last name: _____
- Date: _____ Email address: _____
- Grading choice: ☐ pass/no-pass ☐ grade
(choose one, you may change your mind later and resubmit this survey)

Computer Background

- Previous computer classes or training taken:

- Work or other experience using computers:

Study Groups

Students often like to work together on assignments and prepare for tests. However you may not know anyone else in the class to work with.

- Would you like to participate in a CIS 90 study group? ☐ yes ☐ no
- If so:
 - Would you like to participate: ☐ face-to-face ☐ online ☐ either way
 - Would you like the instructor to help place you in a study group with other interested classmates? ☐ yes ☐ no

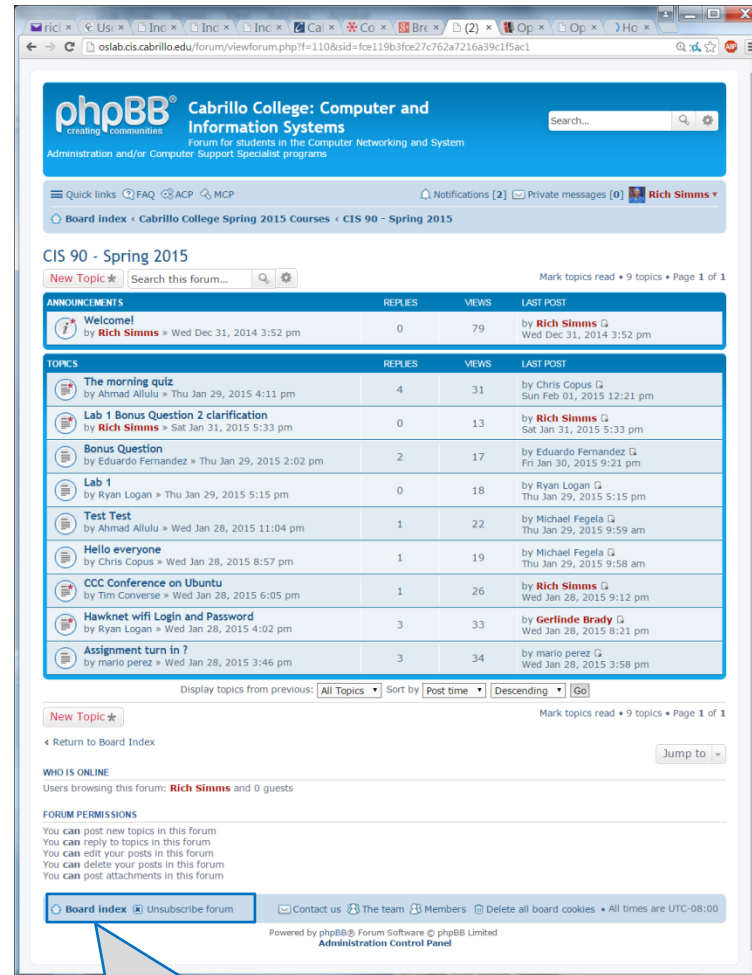
Course Objectives

- What are you hoping to learn in this class?

- Other comments or special learning needs?

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

To get notifications of new forum posts



1) Login to the forum

2) Go to the CIS 90 forum

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

This is what it should look like

[Board index](#) [Unsubscribe forum](#)

Help Available in the CIS Lab

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



CIS 90 Student Lab Assistants



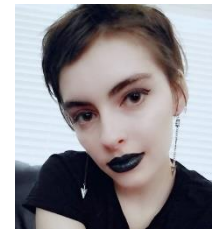
Kevin

T/Th 1:30-4:30
W 12:30-4:30



Roberto

T/Th 3:30-7:30



Tess

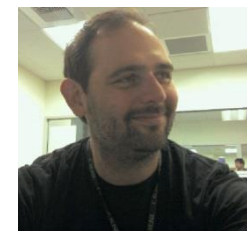
T/F 2-6:30
W/Th 12-5

Linux Instructors



Rich Simms

M 9-11:30



Mike Matera

Rich's Cabrillo College CIS Classes Home Page

Home

Resources

Forums

CIS Lab

Canvas

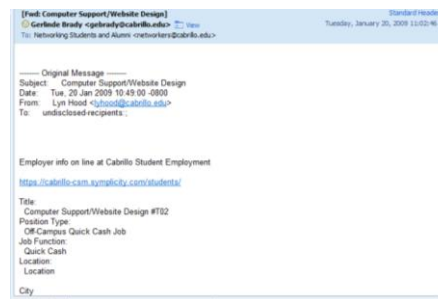
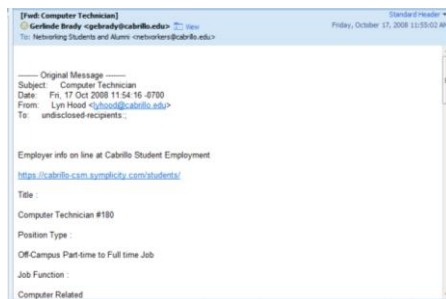
Note, Carter, a CIS 90 alumnus, is volunteering this term to help CIS 90 students with questions. His hours are on the white board in the CIS Lab.

Cabrillo Networking Program Mailing list

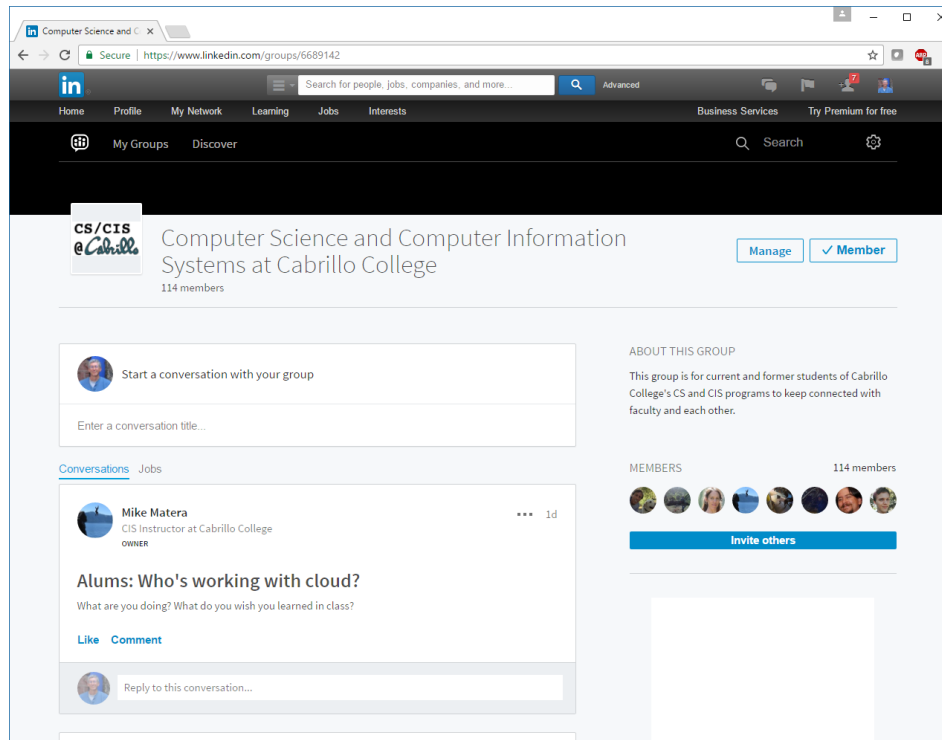
Subscribe by sending an email (no subject or body) to:

networkers-subscribe@cabrillo.edu

- Program information
- Certification information
- Career and job information
- Short-term classes, events, lectures, tours, etc.
- Surveys
- Networking info and links



LinkedIn Computer Science and Computer Information Systems at Cabrillo College

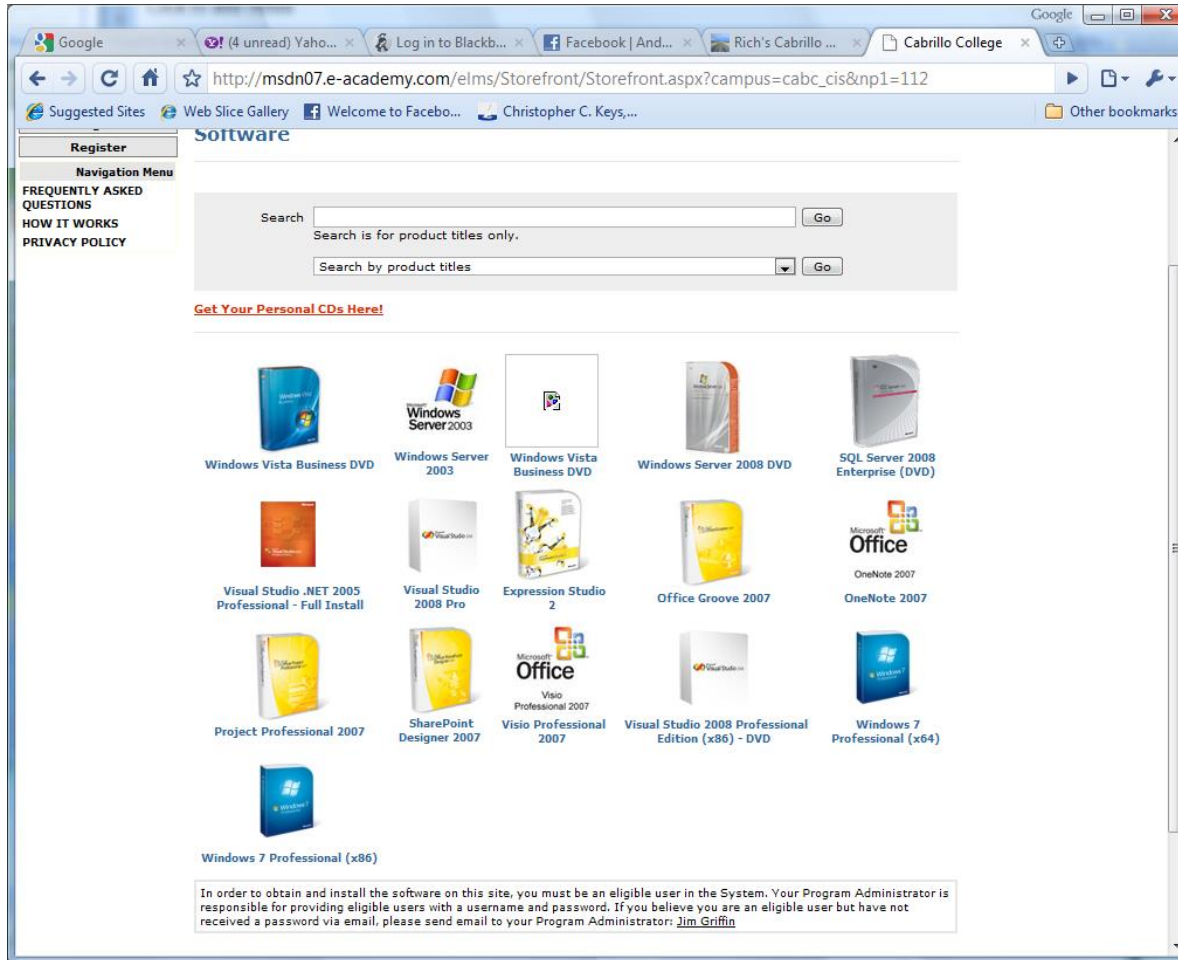


For 3 points extra credit:

- 1) Join LinkedIn.com
- 2) Join this group
- 3) Send me an email when finished.

<https://www.linkedin.com/groups/6689142>

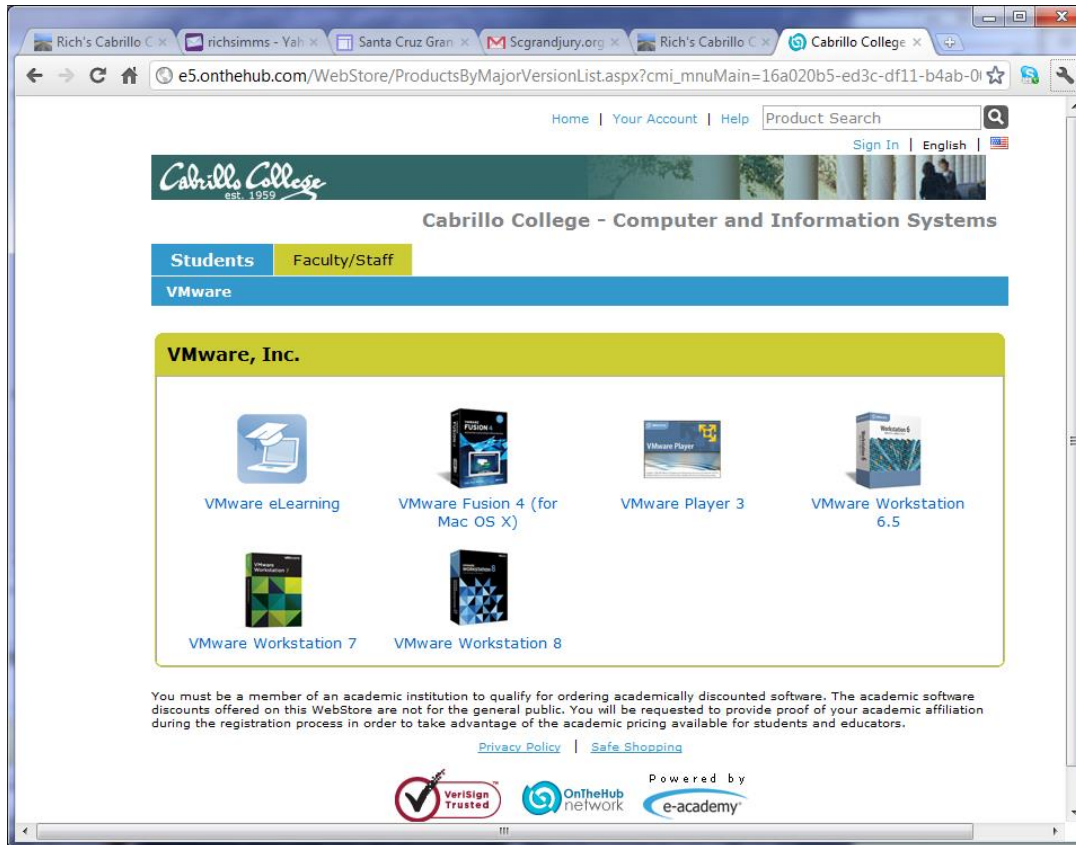
MSDN Academic Alliance



- Microsoft software for students registered in a CIS or CS class at Cabrillo
- Available after registration is final (two weeks after first class)

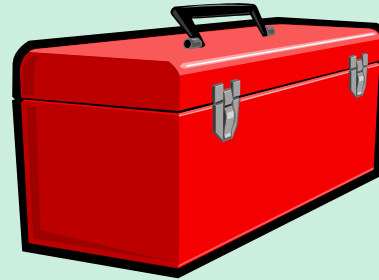
To get to this page, go to **<http://simms-teach.com/resources>** and click on the appropriate link in the Tools and Software section

VMware e-academy



- VMware software for students registered in a CIS or CS class at Cabrillo
- Available after registration is final (two weeks after first class)

To get to this page, go to **<http://simms-teach.com/resources>** and click on the appropriate link in the Tools and Software section



Lesson 2

Commands



Lesson 2 commands for your toolbox

echo

- Prints 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

set

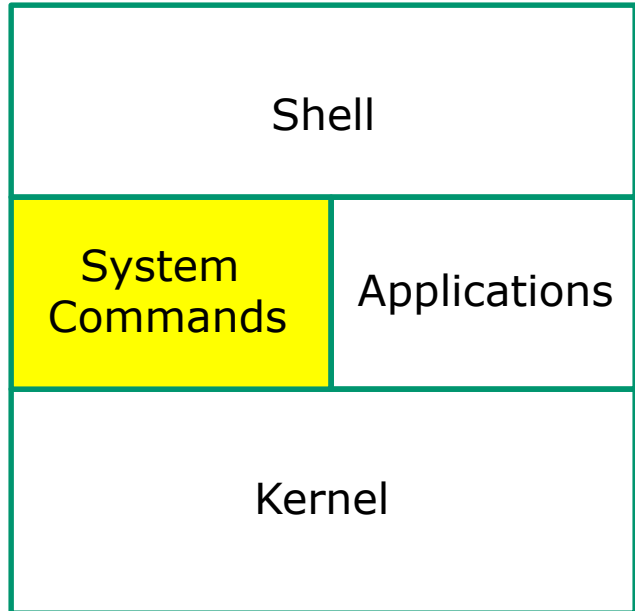
- List all shell variables

env

- List all environment variables

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.



Follow Me

echo
banner

- Prints text and variables
- Make a banner

ls
cat
file
type
apropos
whatis
man
info

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

bc

- Binary calculator

Lesson 2

Commands

Supplemental examples

echo command

Print text and variables

Syntax:

echo *[string]*

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

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


banner command

Output a banner

Syntax:

banner [string]

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

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

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

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

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

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

ls command

List files or directory contents

Syntax:

ls [pathname]

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

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

```
bigfile  Lab2.0      mission    proposal3  text.fxd
bin      Lab2.1      Poems      small_town  timecal
empty    letter      proposal1  spellk      what_am_i
Hidden   Miscellaneous  proposal2  text.err
```

*Listing the contents of
the current directory*

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

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

*Listing the contents of
the Poems directory*

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

Listing three files

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

cat command

Concatenate and view file contents

Syntax:

cat *[pathname]*

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

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

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

< snipped >

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

Alan Sherman

What happens if you spell cat backwards instead? 😊

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

Search for a command on the path

Syntax:

type [command]

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

```
[rsimms@opus run]$ type cal
```

```
cal is /usr/bin/cal
```

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

*name of the file
(command/program)*

*name of the directory
where file is found*

```
[rsimms@oslab ~]$ type bogus
```

```
-bash: type: bogus: not found
```

bogus is not on the user's path

```
[rsimms@opus run]$ type uname cal
```

```
uname is /bin/uname
```

```
cal is /usr/bin/cal
```

*uname is in the **/bin** directory*

*cal is in the **/usr/bin** directory*

```
[rsimms@oslab ~]$ type type
```

```
type is a shell builtin
```

*type is built into the shell
program*

apropos command

search the whatis database for strings

Syntax:

apropos *string*

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

whatis command

search the whatis database for commands

Syntax:

whatis *command*

```
/home/cis90/simben $ whatis echo
```

```
echo (1) - display a line of text
```

```
echo (1p) - write arguments to standard output
```

```
echo [builtins] (1) - bash built-in commands, see bash(1)
```

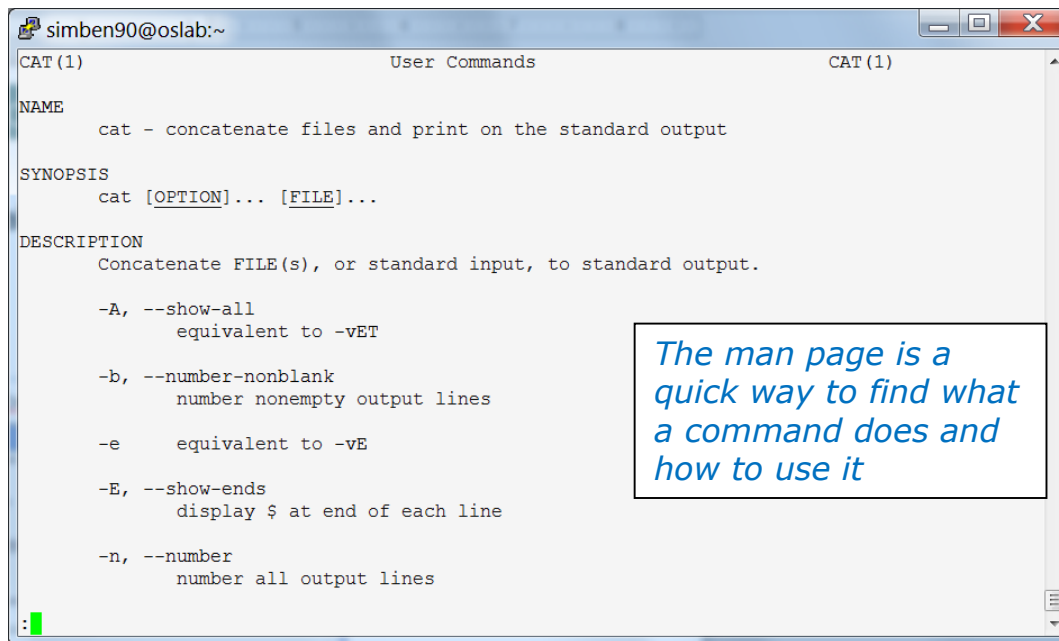
man command

Show the manual page (documentation) for a command

Syntax:

man *command*

/home/cis90/simben \$ **man** **cat**



```

simben90@oslab:~
CAT(1)                                User Commands                                CAT(1)
NAME
    cat - concatenate files and print on the standard output

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

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

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

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

    -e
        equivalent to -vE

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

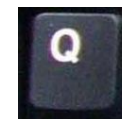
    -n, --number
        number all output lines

:
  
```

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



Use these keys to scroll



Use q key to quit

info command

Alternate documentation tool for commands

Syntax:

info command

Similar to man but has links to additional pages

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

```

simben90@oslab:~
file: bc.info, Node: Top, Next: Introduction, Prev: (dir), Up: (dir)

* Menu:
* Introduction::
* Basic Elements::
* Expressions::
* Statements::
* Functions::
* Examples::
* Readline and Libedit Options
* Comparison with Other Interpreters
* Limits::
* Environment Variables::

--zz-Info: (bc.info.gz)Top,
Welcome to Info version 4.1

simben90@oslab:~
file: bc.info, Node: Examples, Next: Readline and Libedit Options, Prev: Functions, Up: Top
6 Examples
*****
In /bin/sh, the following will assign the value of "pi" to the shell
variable PI.

    pi=$(echo "scale=10; 4*a(1)" | bc -l)

The following is the definition of the exponential function used in
the math library. This function is written in POSIX 'bc'.

    scale = 20

    /* Uses the fact that e^x = (e^(x/2))^2
       When x is small enough, we use the series:
       e^x = 1 + x + x^2/2! + x^3/3! + ...
    */

    define e(x) {
        auto a, d, e, f, i, m, v, z

        /* Check the sign of x. */
        if (x<0) {
            m = 1

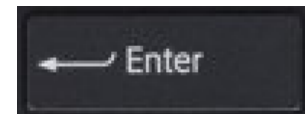
```



Use these keys to scroll



Use q key to quit



Use Enter to follow a link ()*



Use L to go back to last page

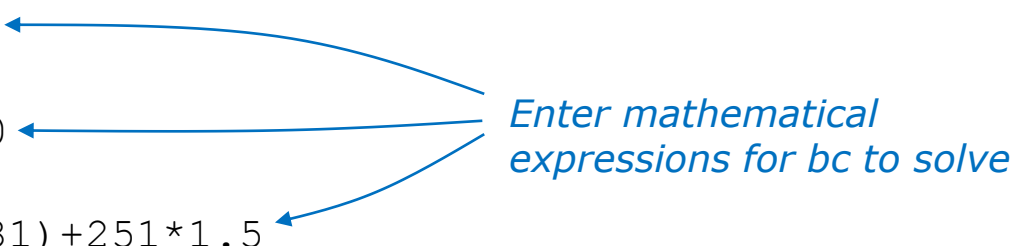
*Move cursor over an * and press Enter to follow link*

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 expressions for bc to solve

*Use quit to
end program*

The 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 or relative pathname must be specified. e.g. **/usr/bin/uname** instead of just **uname**.

Shell Path

The path is used by the shell to locate commands to run

```
/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 colon character is used to separate directories on the path



Locations of common commands

/bin

```

[raimma@server0-01 ~]$
[raimma@server0-01 raimma]$ ls /bin
arch      cut      fgrep    ls       pwd      sync
sh        date    gawk     mail     red      tar
sh_statio dd       grep     more    rm       touch
awk        df       gtar     mktemp  rmdir    true
basename  dmesg   gunzip   mv       rsync    umask
bash       find     iconv    no      sed      undelete_start
ex         gcc      install  su       setfont  undelete_stop
fgrep      echo    ksh       tar      setserial unlink
ftnmdoc   edit    kill      nlcat    sh        usleep
ftnmon    env     link      nlstodomain sleep     vi
ftp        evm     ln         pawk     sort      vxdomainname
uilo       fdisk   lsblk     ping     stty      yfdomainname
x11        fdisk   ps        rm       stat      zcat
[raimma@server0-01 raimma]$

```

Commands for regular users are in /bin and /usr/bin

/usr/bin

[illegible]

/sbin

[illegible]

System administration commands are in /sbin and /usr/sbin

/usr/sbin

```
[isimms@server-01 isimms]$ ls /usr/sbin
```

accept	nmapd
adduser	update
adsl-connect	update
adsl-setup	ntp-generate
adsl-start	nmap
adsl-status	nptimed
adsl-stop	ntpmonst
alternatives	nuptrace
amavisd	ntp-wait
anaconda	atopw
arping	packer
asid	pdblistl
atrun	blat
authconfig	psap_dump
autocount	psap_set
awxapictrl	sage
cisco-activation-keyconf	ppdpump
build-localize-archive	pppoe
build-index-control	pppoe-relay
kernel-lock-helper	pppoe-server
exploit	pppoe-mniff
chkitfontpath	psagets
	prillases

*Most commands reside in these four directories. They can be found in other places as well. For example system administrators often put custom commands in **/usr/local/bin***

The /bin directory

`ls /bin`

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

*/bin has essential
commands used
by everyone.*

*Do you see any of the commands we learned in Lesson 1 in the /bin directory?
Type the names of those commands in the chat window.*

The /usr/bin directory

ls /usr/bin

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

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

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

snipped

```
grotty  
groups  
gs  
gsbj  
gsdj  
gsdj500  
gslj  
gslp  
gsnd  
gsoelim  
gstack  
gst-feedback  
png2theora  
pnm2ppa  
pod2html  
pod2latex  
pod2man  
pod2text  
pod2usage  
podchecker  
podselect  
POST  
post-grohtml  
poweroff  
zforce  
zgrep  
zip  
zipcloak  
zipgrep  
zipinfo  
zipnote  
zipsplit  
zless  
zmore  
znew  
zsoelim  
/home/cis90/simben $
```

Do you see any of the commands we learned in Lesson 1 in the /usr/bin directory? Type the names of those commands in the chat window.

The /sbin directory

```
ls /sbin
```

```

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

```

snipped

```

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

```

```

/home/cis90/simben $

```

These are essential commands and utilities used by system administrators.

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

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

The /usr/sbin directory

```
ls /usr/sbin
```

```

simben90@oslab:~/
/home/cis90/simben $ ls /usr/sbin
abrttd                                hald                                  pwconv
abrt-install-ccpp-hook                htcacheclean                         pwunconv
abrt-server                            httpd                                 quota_nld
accept                                 httpd.event                           quotastats
accton                                 httpd.worker                          raid-check
acpid                                  httxt2dbm                            readprofile
addgnupghome                           hwclock                              redhat_lsb_trigger.i686
adduser                                iconvconfig                           reject
alsactl                                iconvconfig.i686                     repquota
alternatives                           ipa-client-install                   restorecond
anacron                                ipa-getkeytab                         rotatelog
apachectl                              ipa-join                             rpcdebug
applygnupgdefaults                    ipa-rmkeytab                         rpc.gssd
arpd                                    irqbalance                           rpc.idmapd
arping                                 kbd5_send_err                        xzinfo
snipped
getenforce                             postconf                             userhelper
getpcaps                               postdrop                             usermod
getsebool                              postfix                              usernetctl
glibc_post_upgrade.i686                postkick                             vibr
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.

Use the **type** command to find a command on the path

Syntax:

type [command]

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

```
[rsimms@opus run]$ type cal
```

```
cal is /usr/bin/cal
```

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

*name of the file
(command/program)*

*name of the directory
where file is found*

```
[rsimms@oslab ~]$ type bogus
```

```
-bash: type: bogus: not found
```

bugus is not on the user's path

```
[rsimms@opus run]$ type uname cal
```

```
uname is /bin/uname
```

```
cal is /usr/bin/cal
```

*uname is in the **/bin** directory*

*cal is in the **/usr/bin** directory*

```
[rsimms@oslab ~]$ type type
```

```
type is a shell builtin
```

type is built into the shell

Class Activity

- 1) Where are the **scavenge** and **sc** commands?
- 2) What kind of files are they?

Type your answers in the chat window.

Switch to electronic white board

Class Activity

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

bc

/bin

tty

/usr/bin

echo

/sbin

ifconfig

/usr/sbin

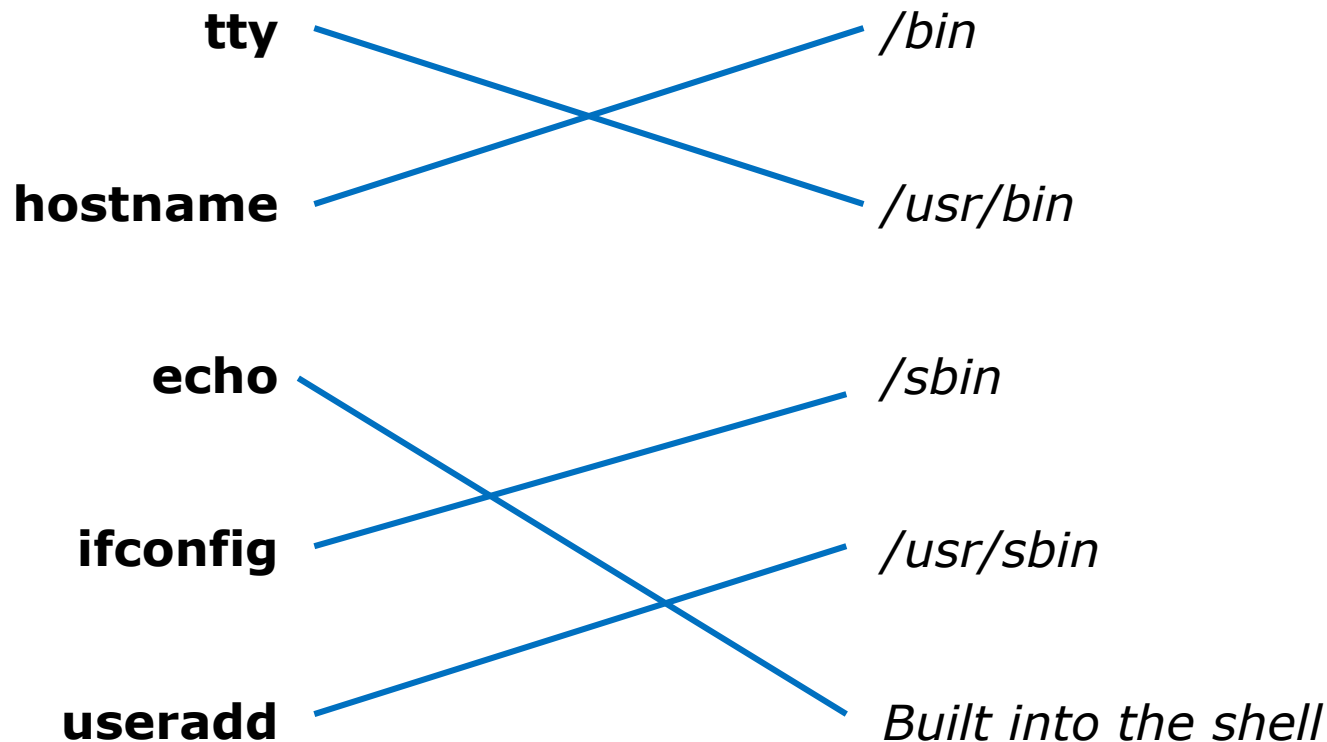
useradd

Built into the shell

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

Class Activity

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



*Switch
back to
slides*



Programs

Binary code
vs text scripts



UNIX commands & utilities are executable programs

A program can be binary code:

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

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.
- Examples: The apropos command.

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

Try both programs (commands)
to see what 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
```


Use **type** to find where the programs are on the path



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.

Use the **ls** command to list the programs files



apropos



cal

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

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

Note, both files show as green because they are executables (more on this later)

```
/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.*

Use the **file** command to get additional info on the files



apropos



cal

```
/home/cis90/simben $ file /usr/bin/apropos  
/usr/bin/apropos: POSIX shell script text executable
```

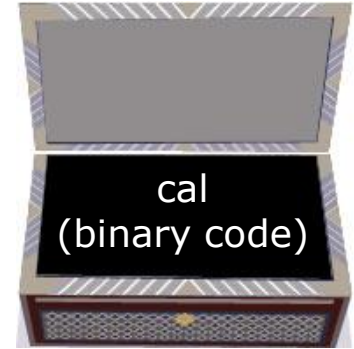
apropos is a shell script

```
/home/cis90/simben $ file /usr/bin/cal  
/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
```

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



Viewing the contents of the program files



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

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

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

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

[illegible]

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

How binary programs are created



cal

From: gcal-3.01.tar.gz

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

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

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

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

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

FYI

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

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

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

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

Hi Folks,

Does anyone happen to know if there are ways to manipulate output from uname such that it is listed in the order that I want it to be? Under 'Commands' in Lab #2, question 11, we are asked what options would we use to display just the operating system, 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?

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

Type your answer in the chat window.

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

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

Type your answer in the chat window.

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

Paste a line of output in the chat window.

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

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

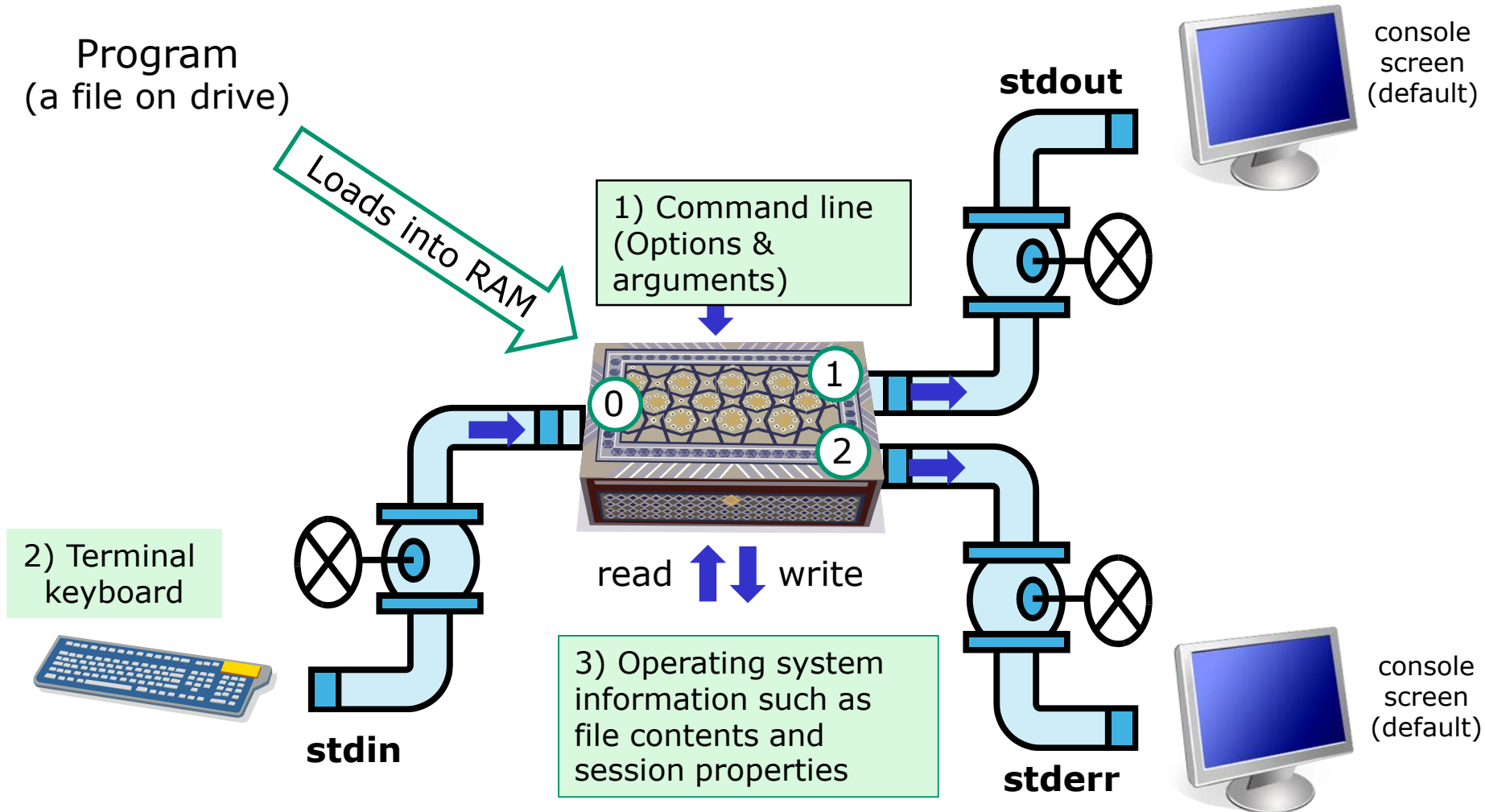
Type your answer in the chat window.

Inputs to Commands

You will get these questions when you submit Lab 2

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

Inputs to Commands



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

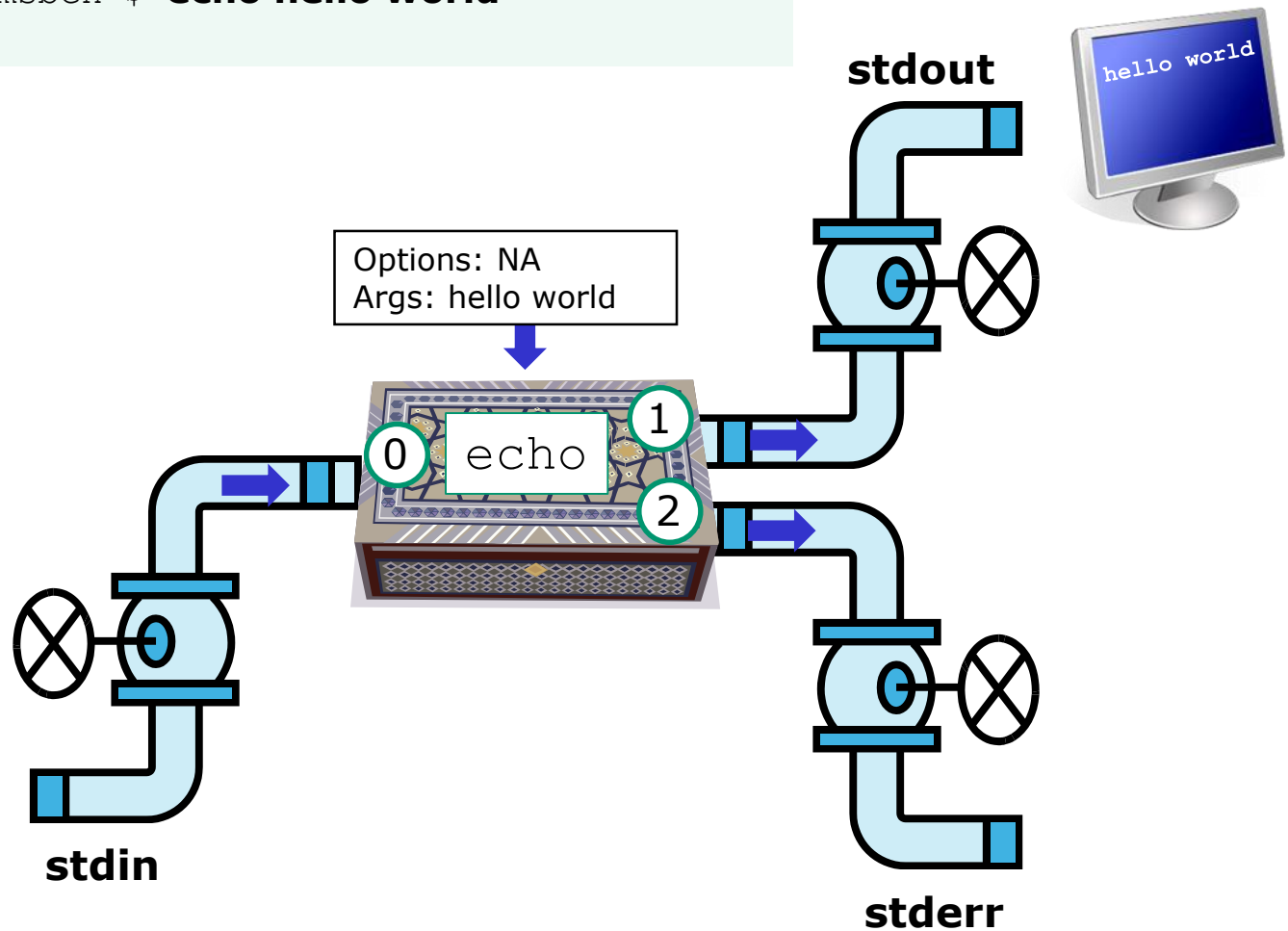
```
/home/cis90/simmen $ echo hello world  
hello world
```

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

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

echo command

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



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

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

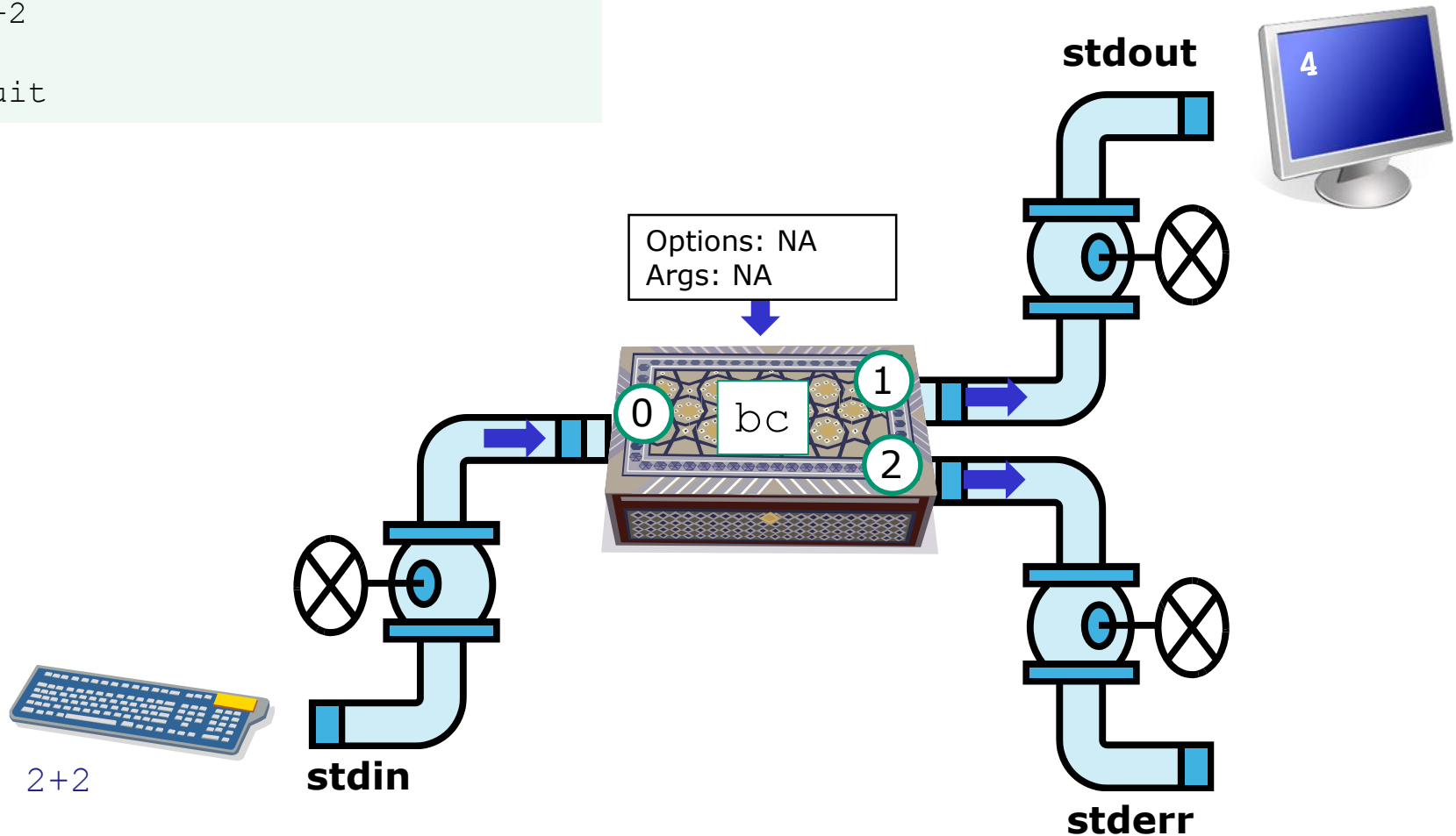
```
/home/cis90/simmsben $ bc
bc 1.06
Copyright 1991-1994, 1997, 1998, 2000 Free
Software Foundation, Inc.
This is free software with ABSOLUTELY NO
WARRANTY.
For details type `warranty'.
2+2
4
500-200+3
303
sqrt(64)
8
quit
```

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

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

bc command

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



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

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

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

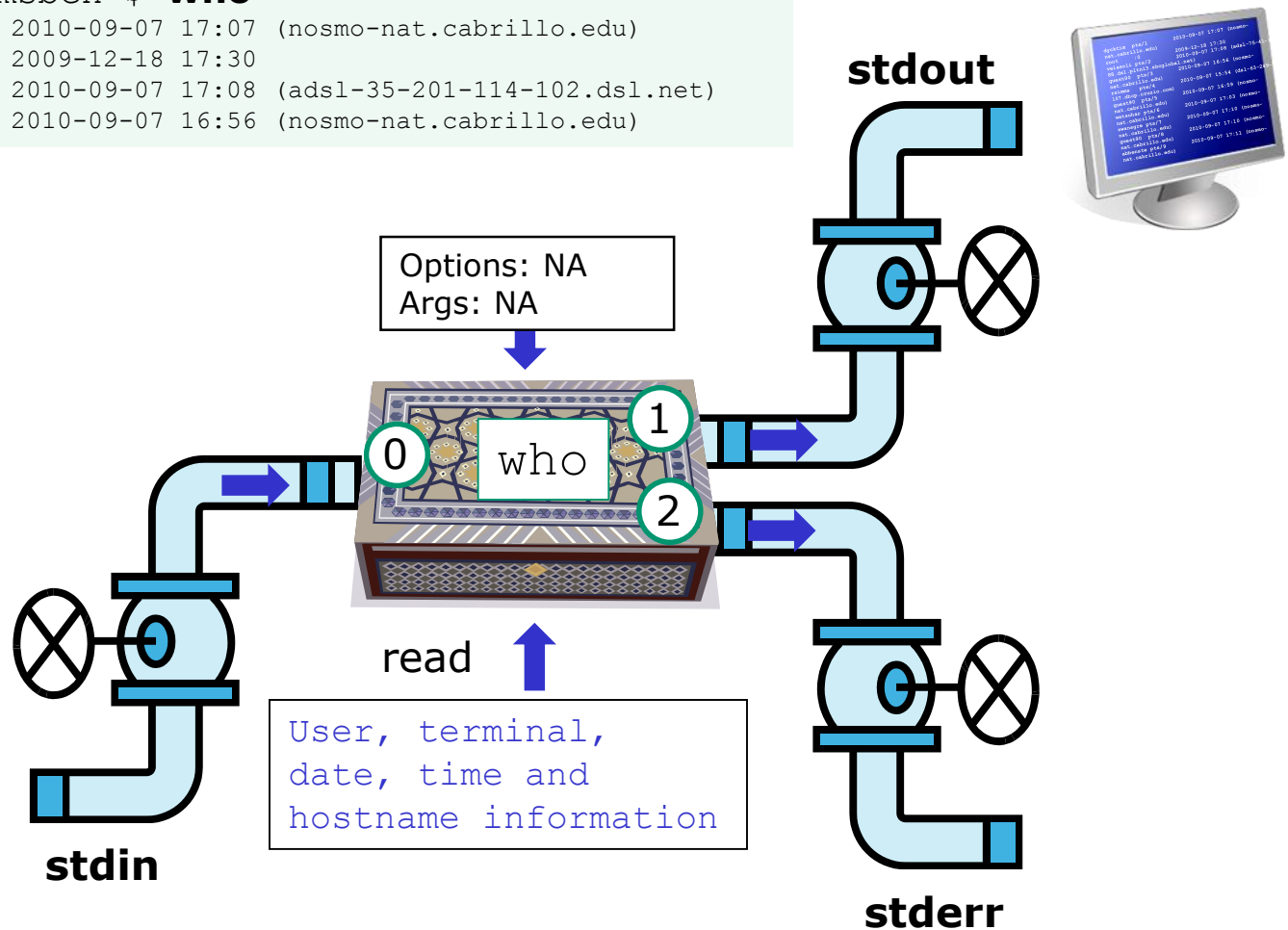
```
/home/cis90/simben $ uname
Linux
```

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

who command

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

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



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

Class Activity

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

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

Type your answer in the chat window

Command Syntax

(grammar lesson)

Some new vocabulary

from Dictionary.com

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

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

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

Command Syntax

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

Command – is the name of an executable program file.

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

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

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

Command Syntax Rules

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

Command – usually at the beginning of the line

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

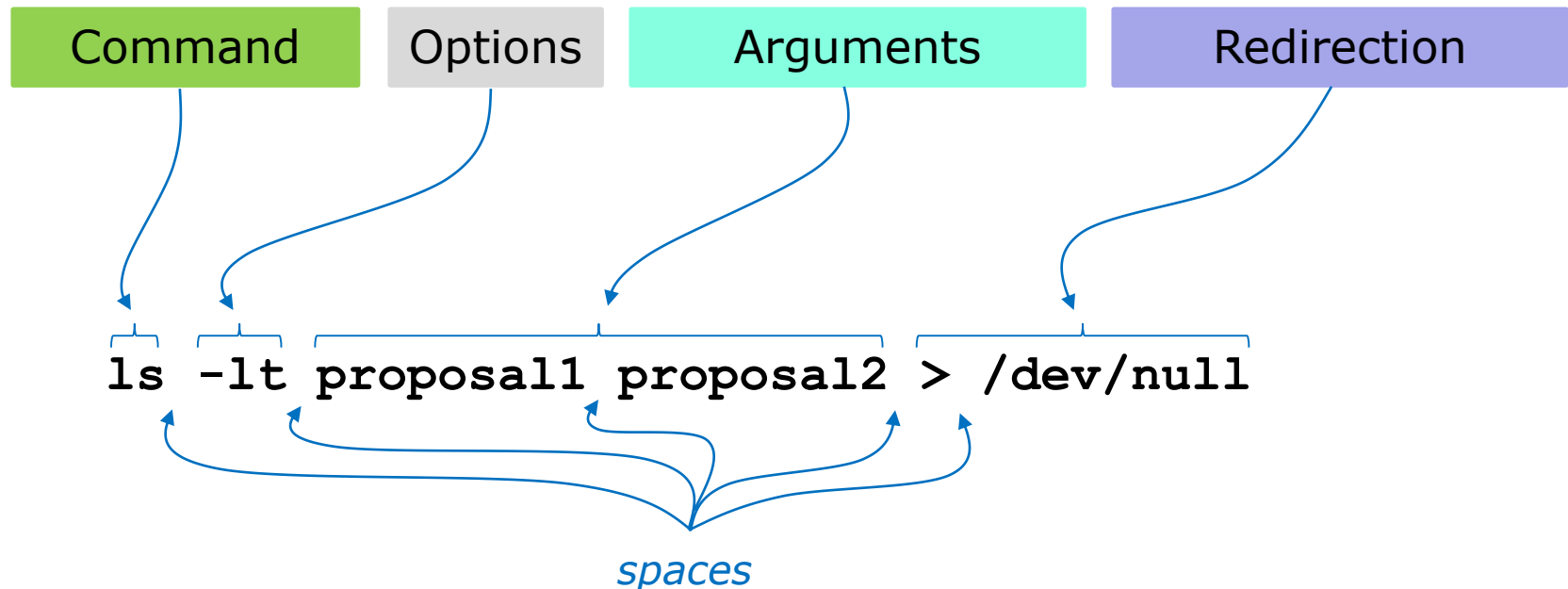
Arguments – follow the options. Multiple arguments must be separated by spaces.

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

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

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

Command Syntax Example



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

More Command Syntax Examples

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

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

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

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

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

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

More on redirection in later lessons

Parsing

Command Syntax

Command

Options

Arguments

Redirection

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

Use the chat window to type your answers

Command:

Options:

How many:

What are they:

Arguments:

How many:

What are they:

Redirection:

How many:

What is redirected:

Command Syntax

Command

Options

Arguments

Redirection

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

Please parse the command line above

Command: echo

Options:

How many: NA
What are they: NA

Arguments:

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

Redirection:

How many: NA
What is redirected: NA

Command Syntax

Command

Options

Arguments

Redirection

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

Use the chat window to type your answers

Command:

Options:

How many:

What are they:

Arguments:

How many:

What are they:

Redirection:

How many:

What is redirected:

Command Syntax

Command

Options

Arguments

Redirection

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

Please parse the command line above

Command: ls

Options:

How many: 2
What are they: l, d

Arguments:

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

Redirection:

How many: NA
What is redirected: NA

Command Syntax

Command

Options

Arguments

Redirection

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

Use the chat window to type your answers

Command:

Options:

How many:

What are they:

Arguments:

How many:

What are they:

Redirection:

How many:

What is redirected:

Command Syntax

Command

Options

Arguments

Redirection

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

Please parse the command line above

Command: ls-ld/bin/usr/bin

Options:

How many:	NA
What are they:	NA

Arguments:

How many:	NA
What are they:	NA

Redirection:

How many:	NA
What is redirected:	NA

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

Command Syntax

Command

Options

Arguments

Redirection

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

Use the chat window to type your answers

Command:

Options:

How many:

What are they:

Arguments:

How many:

What are they:

Redirection:

How many:

What is redirected:

Command Syntax

Command

Options

Arguments

Redirection

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

Please parse the command line above

Command: file

Options:

How many: NA
What are they: NA

Arguments:

How many: 2
What are they: proposal1, timecal

Redirection:

How many: NA
What is redirected: NA

Variables

Shell Variables

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

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

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

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

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

Shell Environment Variables

These variables are automatically set for you when you log in

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

Showing common 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

Note that Terminal type ≠ Terminal device

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

```

simben90@oslab:~
simben90@oslab.cabrillo.edu's password:
Last login: Tue Feb  4 18:56:49 2014 from ec2-54-215-232-67.us-west-1.compute.am
azonaws.com

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

Welcome to Opus
Serving Cabrillo College

Terminal type? [xterm]
Terminal type is xterm.
/home/cis90/simben $ tty
/dev/pts/1
/home/cis90/simben $ echo $TERM
xterm
/home/cis90/simben $
  
```

*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

Setting Variable Values

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

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

*Show the current
terminal type*

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

*Change the terminal
type and display the
new value*

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

*Change the terminal
type back to the
original value*

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

The SHELL variable

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

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

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

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

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

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

Changing the shell prompt

(PS1 variable)

The PS1 variable

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

The PS1 variable defines the shell prompt

Follow Me

```
/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
```

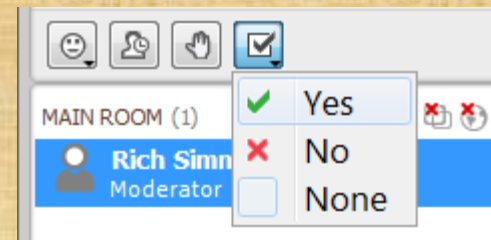
```
What can I do for you simben90?
```

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

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

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

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

More PS1 prompt examples

Changing the prompt

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

\h gets replaced by the hostname

\W gets replaced by the base working directory

\u gets replaced by the username

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

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

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

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

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

user name

hostname

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

indicates regular user

Changing the prompt

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

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

Changing the prompt

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

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



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

Listing the environment variables

Shell (Environment) Variables

env command

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

```
HOSTNAME=oslabs.cabrillo.edu
```

```
SELINUX_ROLE_REQUESTED=
```

```
TERM=xterm
```

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

```
HISTSIZE=1000
```

```
SSH_CLIENT=50.0.68.235 51849 2220
```

```
SELINUX_USE_CURRENT_RANGE=
```

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

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

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

```
USER=simben90
```

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

```
USERNAME=
```

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

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

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

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

```
SELINUX_LEVEL_REQUESTED=
```

```
HISTCONTROL=ignoredups
```

```
SHLVL=1
```

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

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

```
LOGNAME=simben90
```

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

```
CVS_RSH=ssh
```

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

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

```
G_BROKEN_FILENAMES=1
```

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

```
OLDPWD=/bin
```

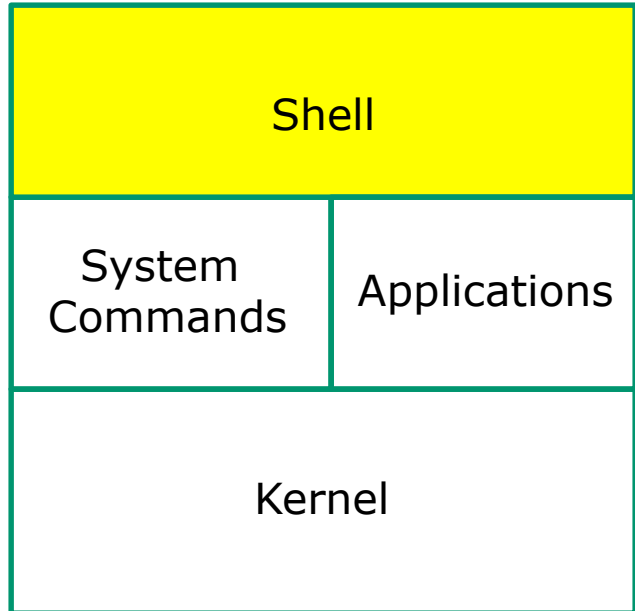
```
/home/cis90/simben $
```

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



The Shell (six steps)

The Shell

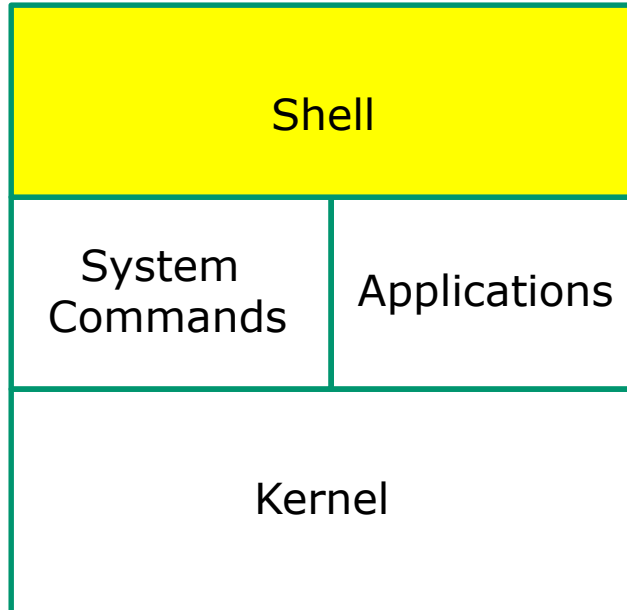


- Allows users to interact with the computer via a "**command line**".
- **Prompts** for a command, parses the command, finds the right program and gets that program executed.
- Is called a "**shell**" because it hides the underlying operating system.
- Multiple shell programs are available: **sh** (Bourne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





Life of the Shell



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





Life of the Shell

Example:

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

Shell Steps

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

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

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



Life of the Shell

Shell Steps

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

1) Prompt user for a command

Example:

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

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

Then you type the command

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

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

```
$PWD $
```

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

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

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



Life of the Shell

2) Parse command user typed

Shell Steps

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

Example:

```
ls -lt proposal1 proposal2
```

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

- Command = ls
- 2 Options = l, t
- 2 Arguments = proposal1, proposal2
- No Redirection



Life of the Shell

3) Search path for the program to run

Shell Steps

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

ls -lt proposal1 proposal2

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

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

*The shell will search each directory in order for an **ls** command*

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

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

Try mimicking what the shell does to search for ls:

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

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

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



Life of the Shell

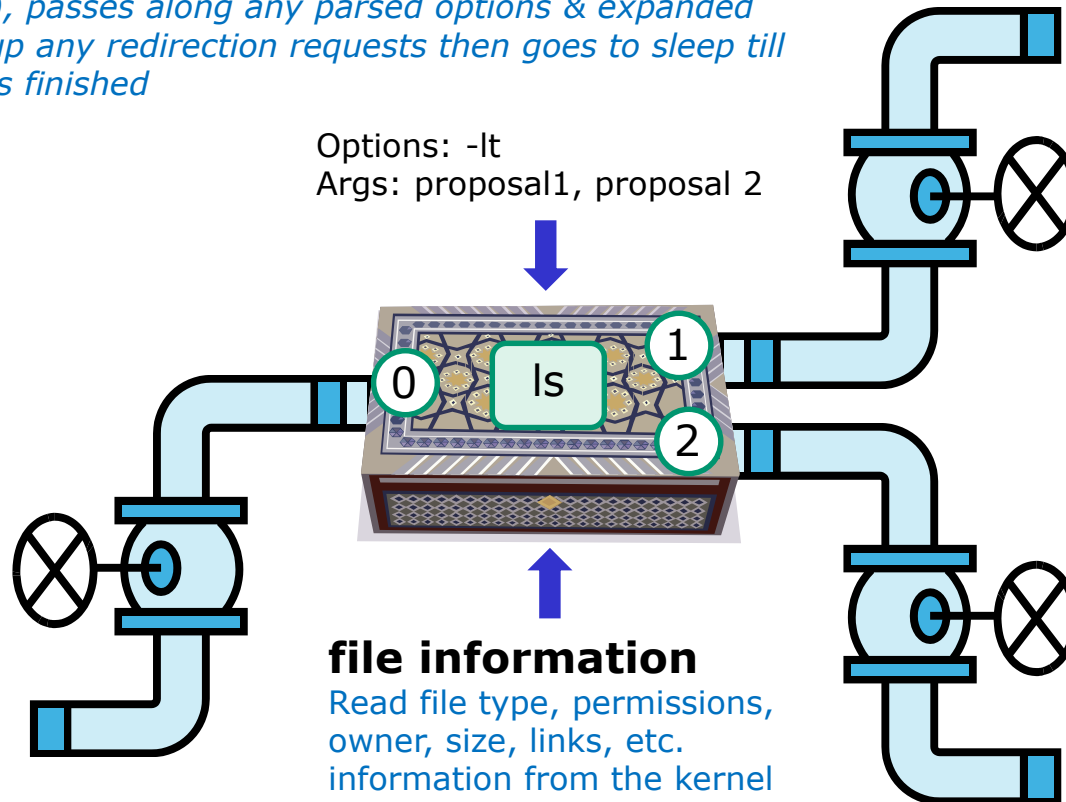
Shell Steps

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

4) Execute the command

```
ls -lt proposal1 proposal2
```

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





Life of the Shell

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

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

Shell Steps

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

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

The shell sleeps while the ls process outputs these two lines



Life of the Shell

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

Shell Steps

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



Life of the Shell

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

-bash: Ls: command not found

What's wrong?

Who output the error?

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

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

What's wrong?

Who output the error?

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

ls: invalid line width: proposal1

What's wrong?

Who output the error?

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

ls: cannot access proposal1proposal2: No such file or directory

What's wrong?

Who output the error?

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

-bash: ls-lt: command not found

What's wrong?

Who output the error?

Meta- characters

Metacharacters

When parsing, the shell gives special meaning to metacharacters

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

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

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

;- allows multiple commands on one line

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

= - use to set variables to new values

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

Other metacharacters we will learn about later include:

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

Metacharacters - quotes

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

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

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

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

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

Metacharacters - quotes

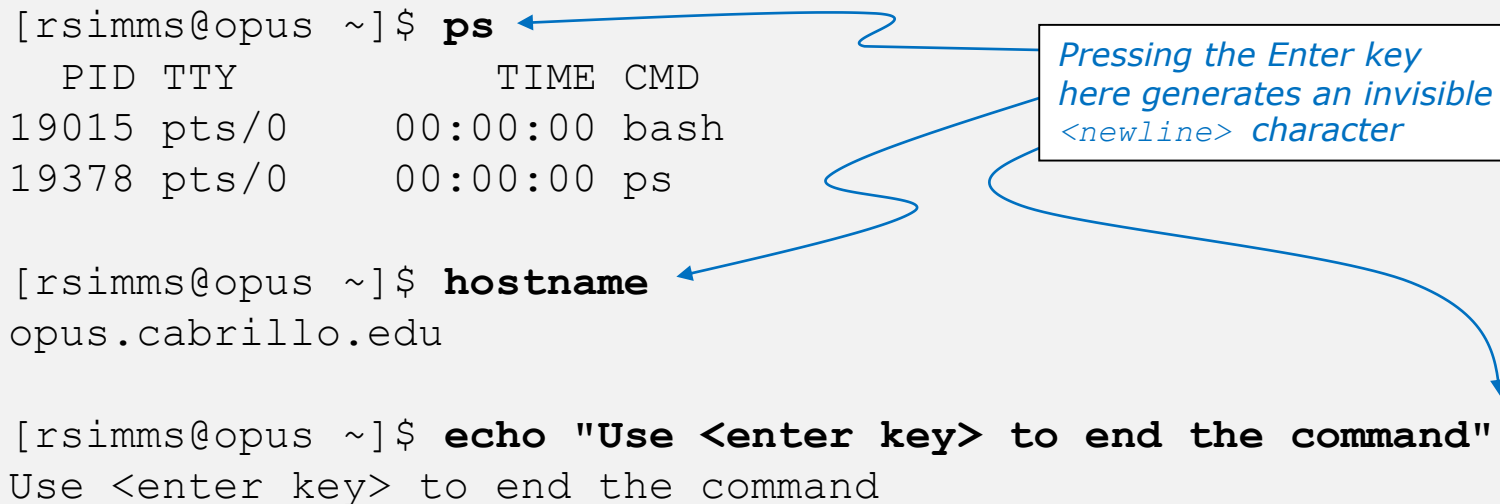
```
/home/cis90/simben $ echo '"double quotes"'  
"double quotes"
```

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

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

Metacharacters - <enter key>

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



The image shows a terminal window with three commands: `ps`, `hostname`, and `echo "Use <enter key> to end the command"`. Blue arrows point from a text box to the end of each command line. The text box states: "Pressing the Enter key here generates an invisible <newline> character".

```
[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
```

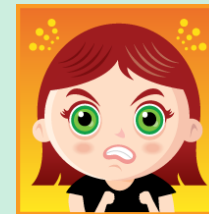
*Press <tab> after the P and B
and the shell fills in the rest*

*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 or relative 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



/usr/bin



/bin

Experiment – Breaking the Path

*Default
path*

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

TROUBLE!

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

/home/cis90/simben $
```

*Break the path by
setting it to null*

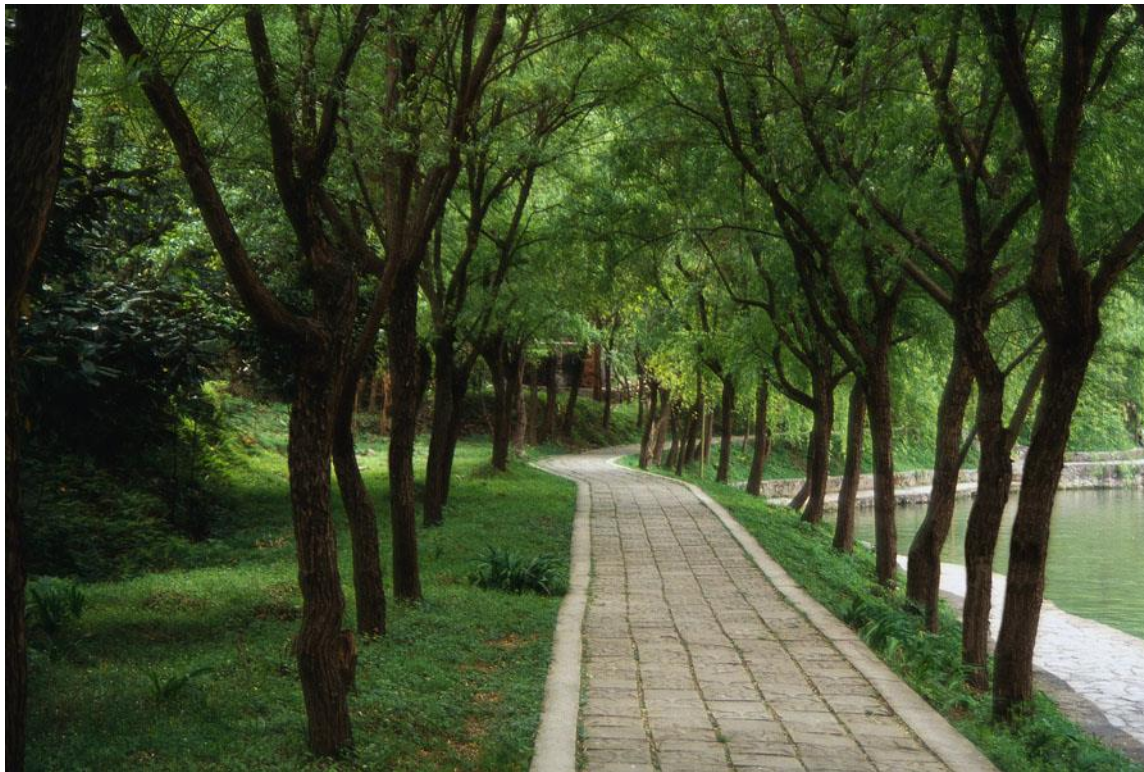
No path

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

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

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

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



There is nothing on the path!

Experiment – Restoring the Path

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

*Add the /bin
directory to the path*

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

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

*echo works
because it is built
into the shell*

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


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



Experiment – Restoring the Path

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

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

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

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

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

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





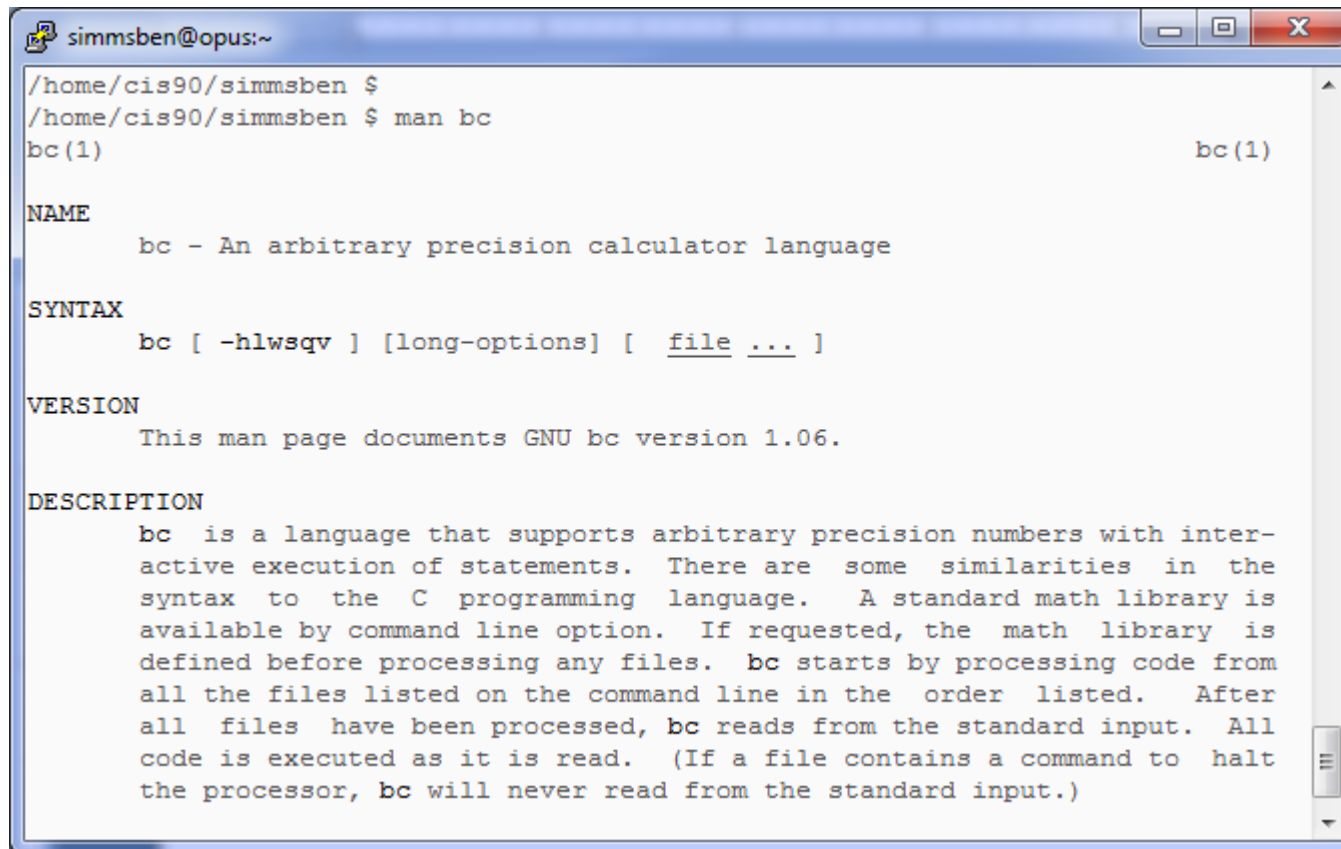
*Need a fresh start -- just log out
and back in again and your path
will be back to normal!*

Docs

Using man (manual) pages

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

Example: **man bc**



```

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

NAME
    bc - An arbitrary precision calculator language

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

VERSION
    This man page documents GNU bc version 1.06.

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



Use these
keys to scroll



Use q key to
quit

Using Google

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

Example: **google** linux bc command

The image shows two overlapping browser windows. The background window is a Google search results page for the query "linux bc command". It shows approximately 1,180,000 results. The foreground window is the "linux.about.com" page for the "bc" command. The page includes a PayPal advertisement at the top, a search bar with the query "linux bc command", and a table of contents with sections for NAME, SYNTAX, and DESCRIPTION. The DESCRIPTION section explains that "bc" is a language for arbitrary precision numbers.

Google Search Results (Background Window):

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

linux.about.com Page (Foreground Window):

- URL: http://linux.about.com/od/commands/l/blcmdl1_bc.htm
- Search query: linux bc command
- Section: **Linux / Unix Command: bc**
- Table of Contents:
 - 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
- Dyn
- John
- MS
- All
- Qu
- su
- Tri
- Vir
- VM
- Wi

Howtos

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

Google | 0 unr... | Yahoo... | Rich's... | The Li... | Cabril... | linux L... |

http://tldp.org/

The Linux Documentation Project

2010-09-06

LDP Worldwide

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

LDP Information

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

Workshop

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

Documents

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

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

FAQs: Frequently Asked Questions
latest updates / main index

man pages: help on individual commands (20060810)

Search / Resources

Links
OMF search

Google | 0 unr... | Yahoo... | 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 mount Command](#) page added.
- September 20: [The head Command](#) page updated.

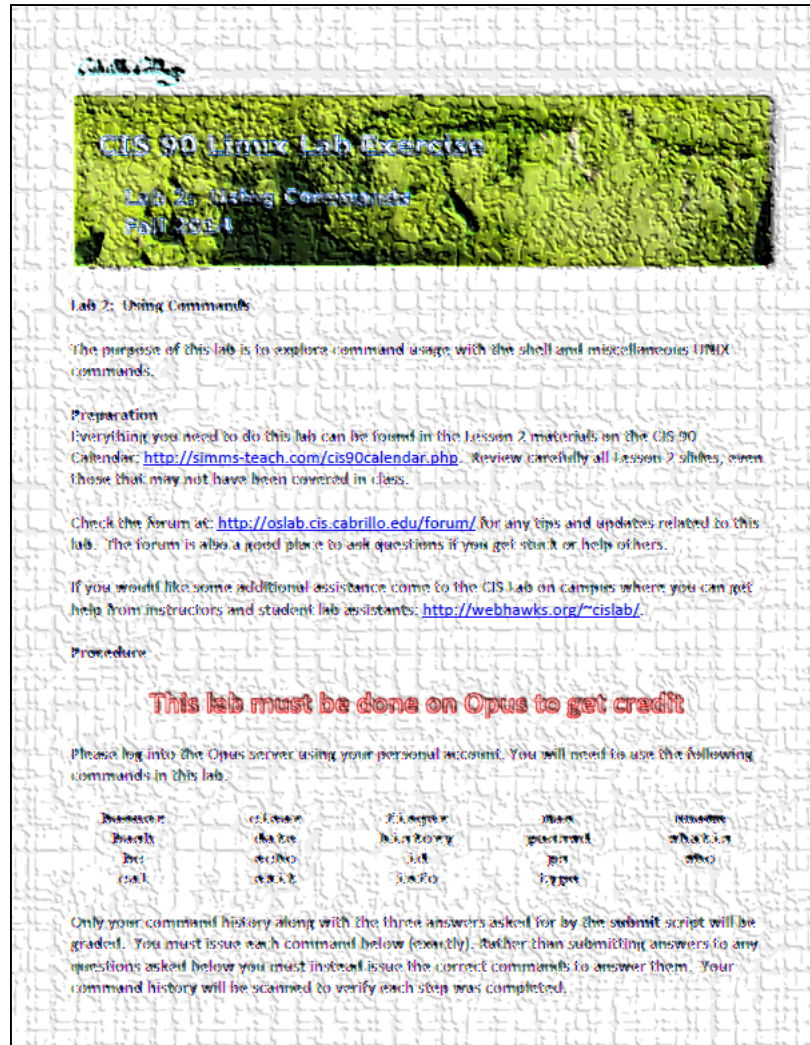
Site Contents:

The Linux Documentation and Information Projects

Assignment



Lab 2 - Using Commands



CIS 90 Linux Lab Description
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:

Description	cd /usr/bin	cd /usr/sbin	man	rm
ls	ls	ls	man	rm
cp	cp	cp	man	rm
mv	mv	mv	man	rm
find	find	find	man	rm

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.

A full-page background image showing a sunset over a beach. The sky is filled with vibrant orange, pink, and purple clouds. The sun is low on the horizon, casting a warm glow. To the right, a dark, silhouetted cliff rises from the beach. The foreground shows the wet sand of the beach reflecting the colors of the sky, with some dark rocks scattered about.

Wrap up

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.

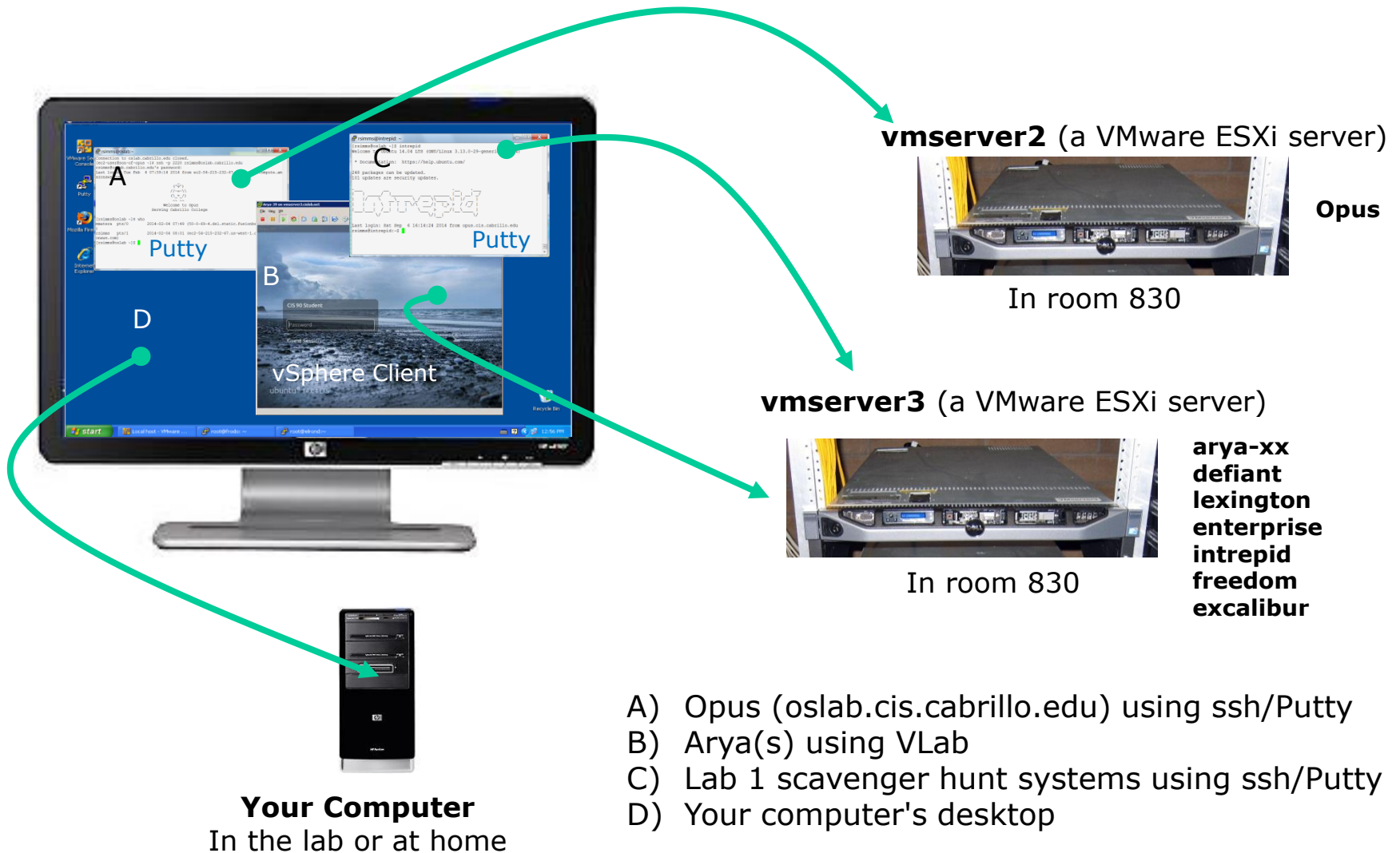
Lab #2

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



FYI

CIS 90 and Smartphones (Android)



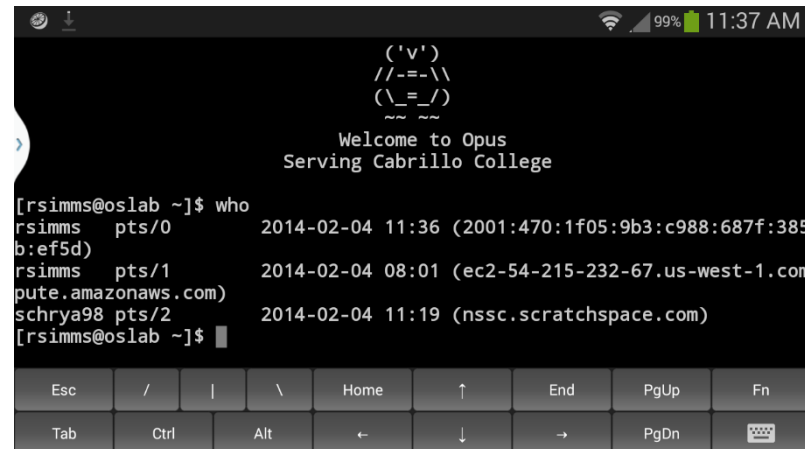
Blackboard
Collaborate App



*Join CCC Confer
virtual classroom*



JuiceSSH - SSH Client app

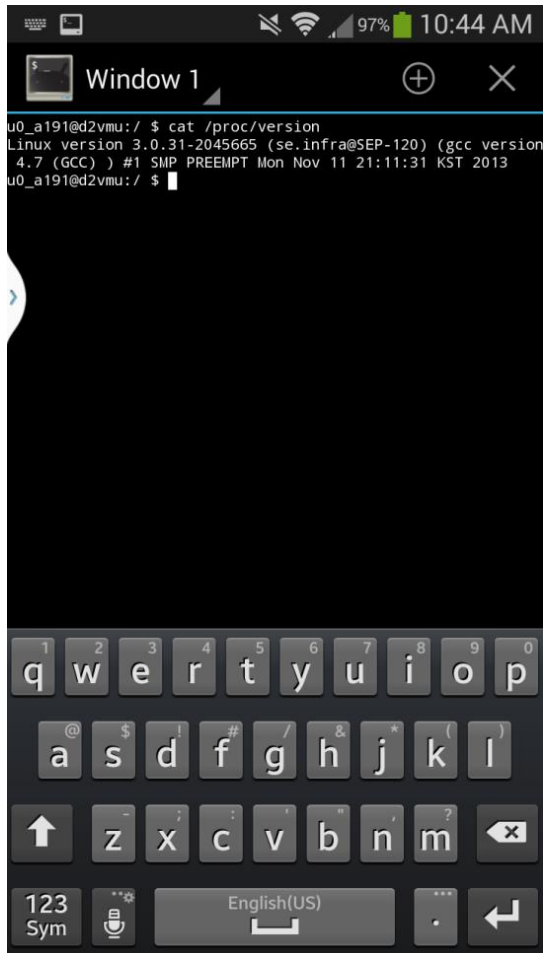


Login to to Opus

CIS 90 and Smartphones (Android)



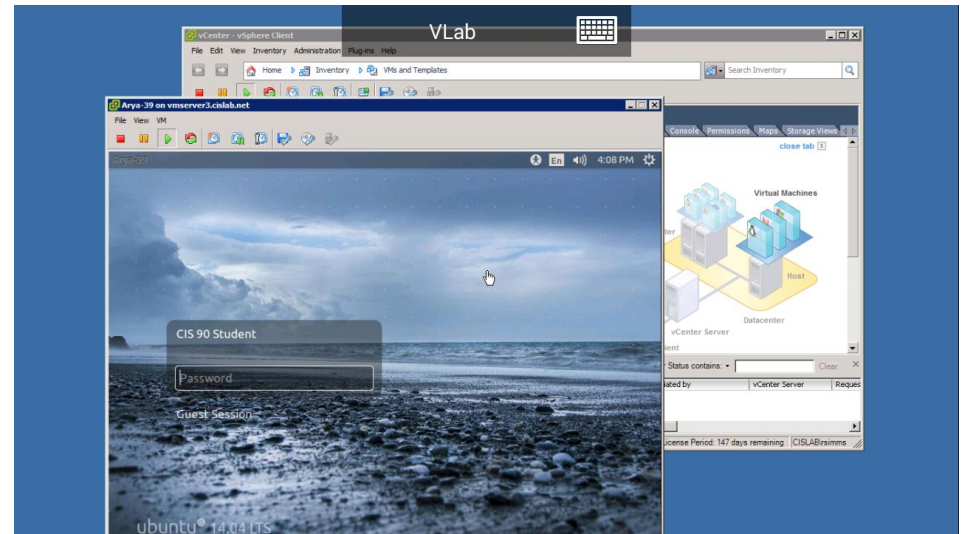
Android Terminal App



Viewing kernel version on
smartphone



Microsoft RDP App



Running Arya VM in VLab

Terminals

Hardware Terminals



Teletype (TTY)



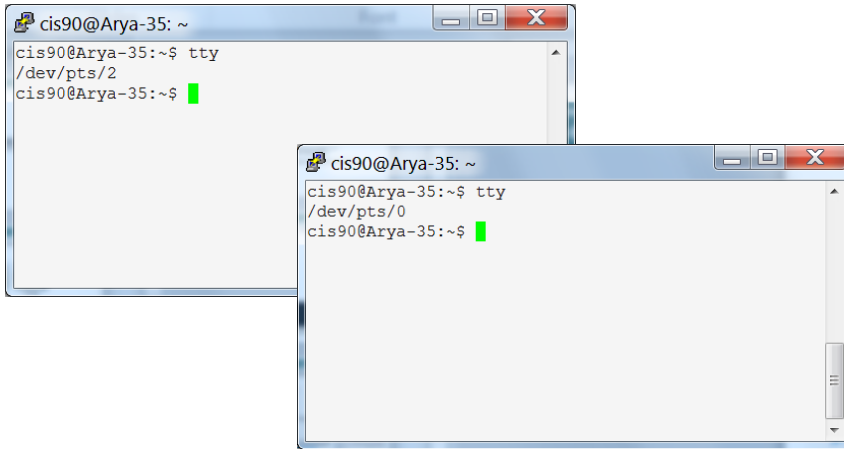
VT100



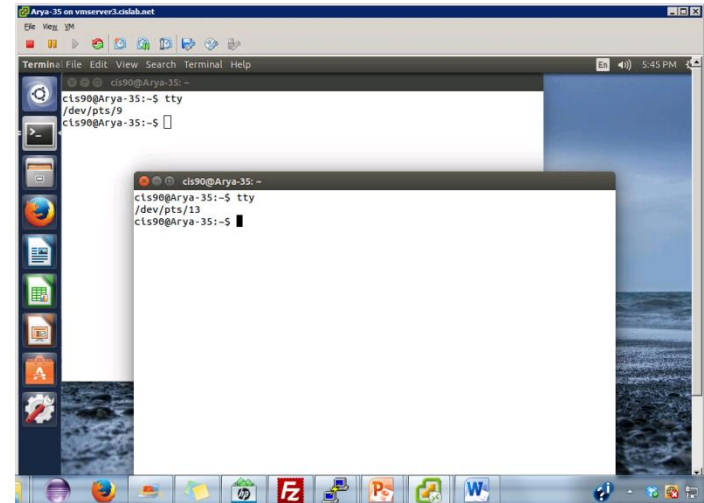
Terminals were used in the old days to interact with "minicomputers" and "mainframe" computers.

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

Software Terminals



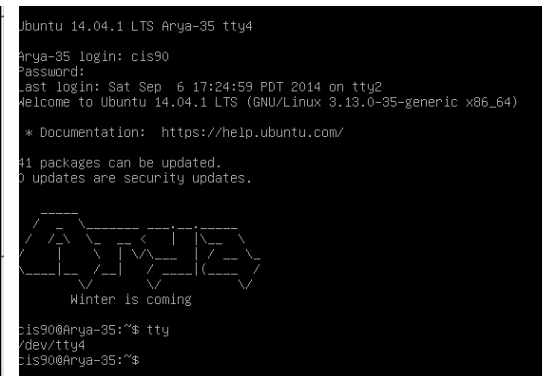
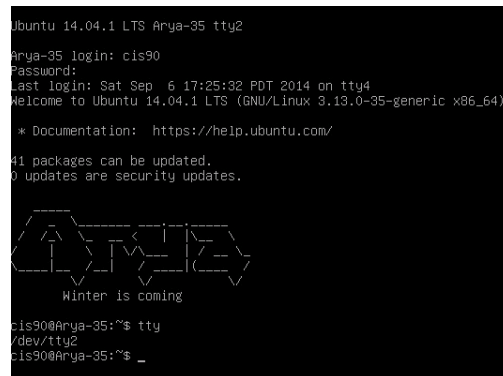
Terminal emulators like PuTTY (with scroll bars, colors, customizable backgrounds, fonts and sizes) for Windows



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

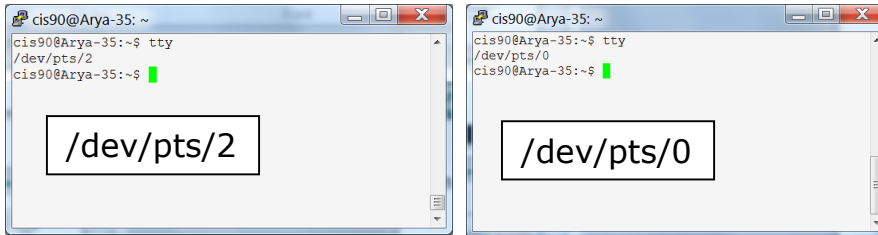
Virtual terminals (use ctrl-alt-fn)

Bare bones, no scroll bars,
also called a console



Various terminal devices on an Arya VM

Terminal emulators (e.g. Putty)



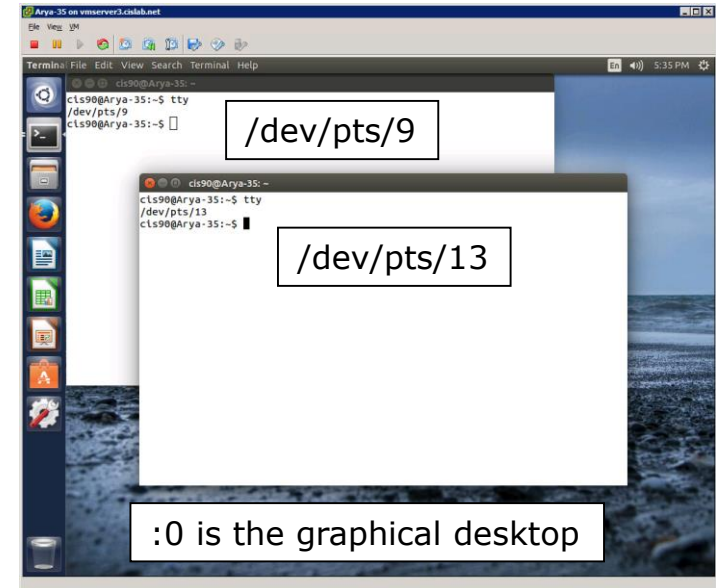
```
cis90@Arya-35:~$ who
cis90    tty4      2014-09-06 17:25
cis90    tty2      2014-09-06 17:25
cis90    pts/2      2014-09-06 17:20 (enterprise.cis.cabrillo.edu)
cis90    :0         2014-09-06 17:20 (:0)
cis90    pts/0      2014-09-06 17:21 (2601:9:6680:53b:4d09:e2b6:e7fc:d999)
cis90    pts/9     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

pts=pseudo terminal,

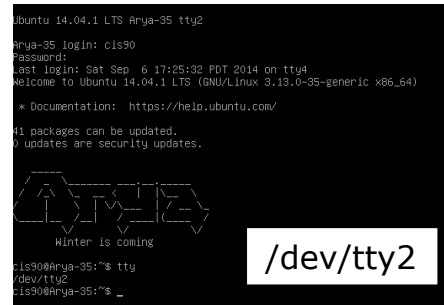
tty=teletype

:n=an X window display number

Graphical terminals on graphical desktop



Virtual terminals



Putty Tips

(Note: tty = teletype)

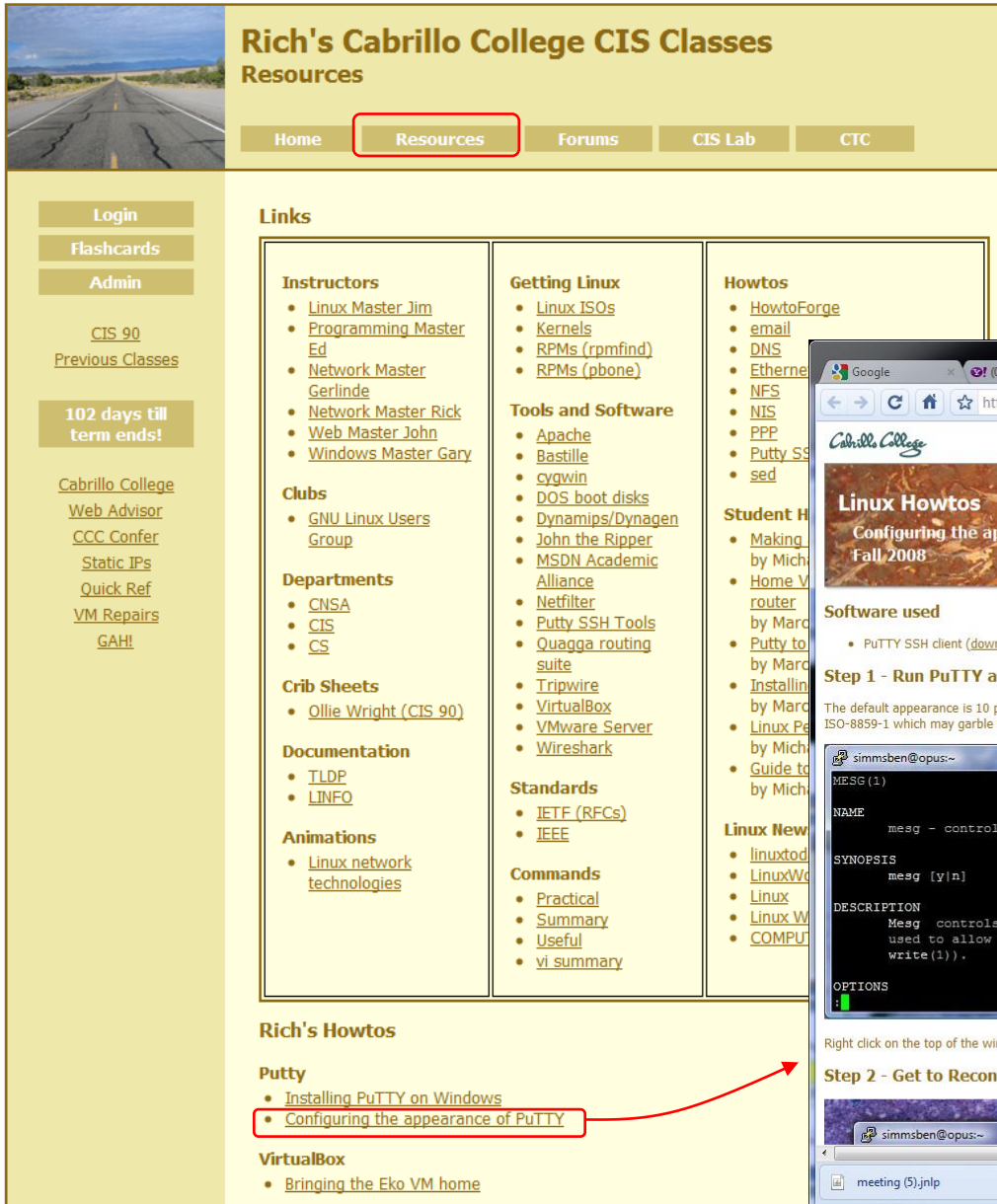
The Putty program

```

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

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

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



Rich's Cabrillo College CIS Classes Resources

Home **Resources** Forums CIS Lab CTC

Login
Flashcards
Admin

CIS 90
Previous Classes

102 days till term ends!

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

Links

Instructors - Linux Master Jim - Programming Master Ed - Network Master Gerlinde - Network Master Rick - Web Master John - Windows Master Gary 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 - Ethernet - NFS - NIS - PPP - Putty SSH - sed Student Help - Making a backup by Michael - Home V router by Marco - Putty to by Marco - Installing by Marco - Linux Pe by Michael - Guide to by Michael Linux News - linuxtoday - LinuxWorld - Linux - Linux W - COMPU
---	--	--

Rich's Howtos

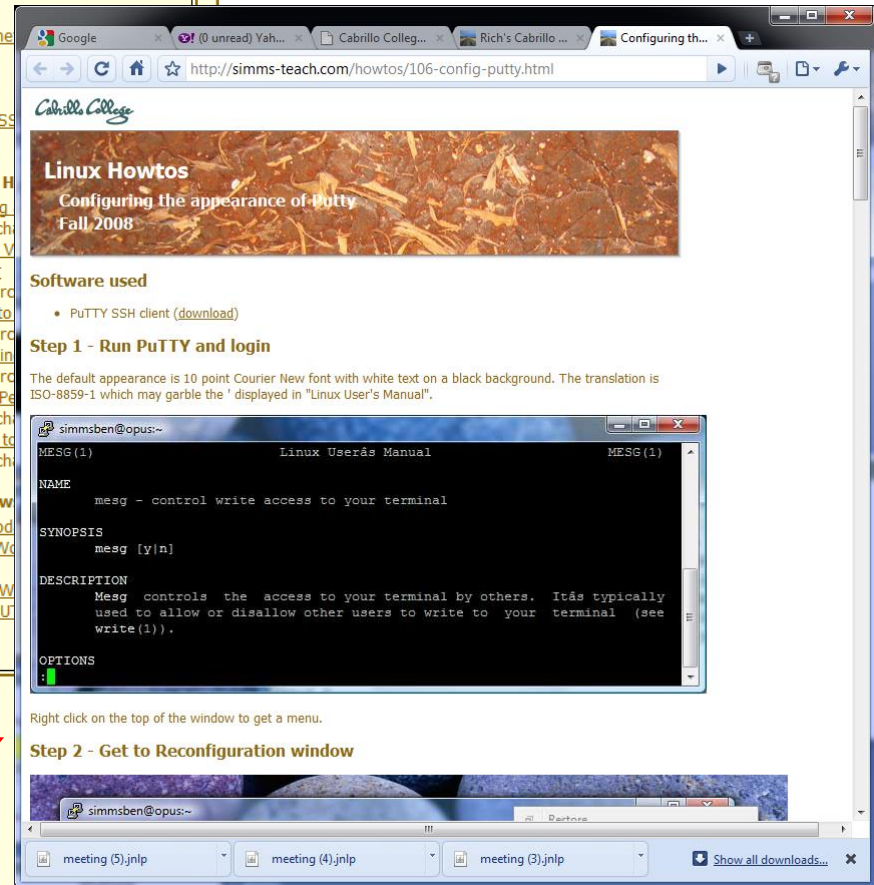
Putty

- Installing PuTTY on Windows
- Configuring the appearance of PuTTY

VirtualBox

- Bringing the Eko VM home

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



Linux Howtos
Configuring the appearance of PuTTY
Fall, 2008

Software used

- PuTTY SSH client (download)

Step 1 - Run PuTTY and login

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

simmsben@opus:~
MSG (1) Linux User's Manual MSG (1)

NAME
msg - control write access to your terminal

SYNOPSIS
msg [y/n]

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

OPTIONS

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

Step 2 - Get to Reconfiguration window

simmsben@opus:~



Lesson 1 Review

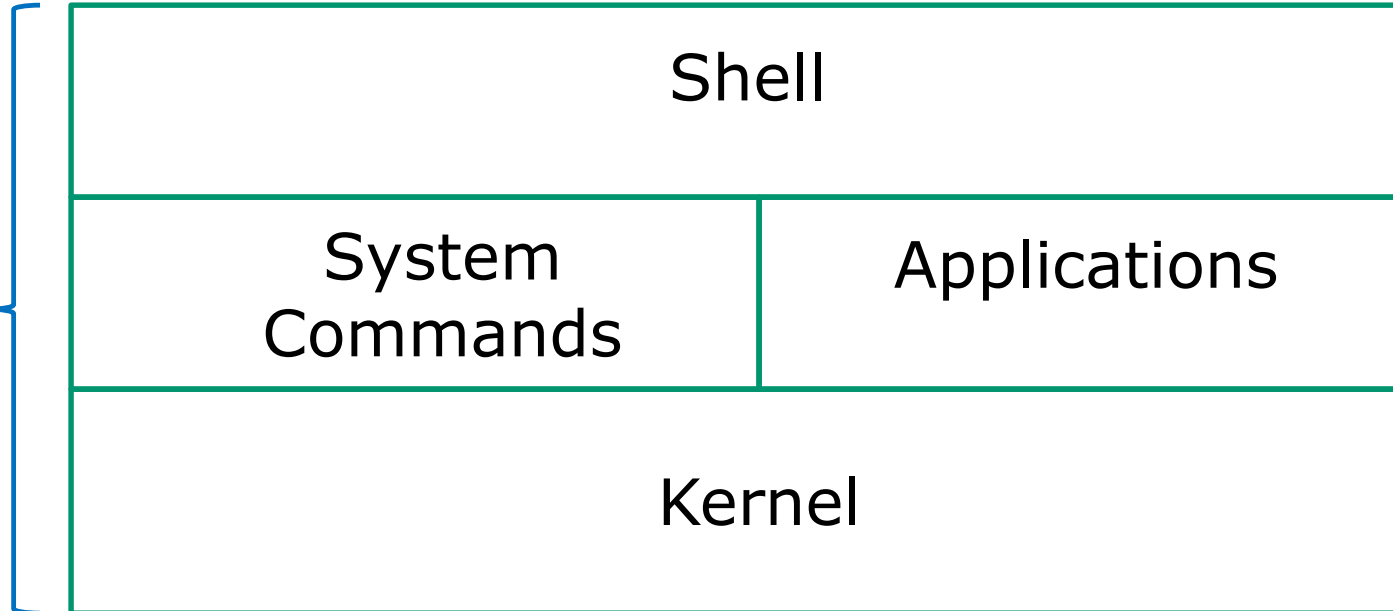
UNIX/Linux Architecture

Simplified View - Four Major Components

Users



Software



Hardware





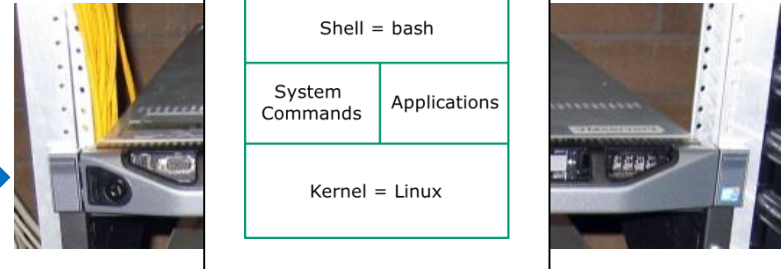
The Lesson 1 commands for your toolbox

cal	<i>Prints calendars</i>
date	<i>Shows the time and date</i>
clear	<i>Clears the screen</i>
exit	<i>Exits login session</i>
history	<i>Shows commands used previously</i>
id	<i>Shows your username and UID (and more)</i>
ps	<i>Shows your processes (including the name of the shell)</i>
ssh	<i>For connecting and logging into a remote computer</i>
hostname	<i>Shows the name of the <u>computer</u> being used</i>
uname	<i>Shows name of the operating system <u>kernel</u></i>
cat /etc/issue	<i>Shows name of the "<u>distro</u>" (distribution)</i>
tty	<i>Shows which terminal device is being used</i>
who	<i>Shows all users who are logged in and from where</i>
who am i	<i>Like who, but only shows your login session</i>

"Name" Terminology



ssh -p 2220 simben90@oslab.cishawks.net



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

Various "names" bandied about:

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

user**name** = simben90

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

(terminal type = xterm)

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

Name of distro = CentOS

Name of shell = bash

Name of kernel = Linux

To view:

/etc/passwd

id

tty

echo \$TERM

hostname

/etc/issue

ps

uname

Terminals types and devices

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

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

Hit Enter to accept

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

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

*The terminal type is **xterm***

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

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

How can I print a calendar?

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

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

*The **cal** command*

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

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

*Month and year **arguments***

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

A command can have arguments

What is the current time and date?

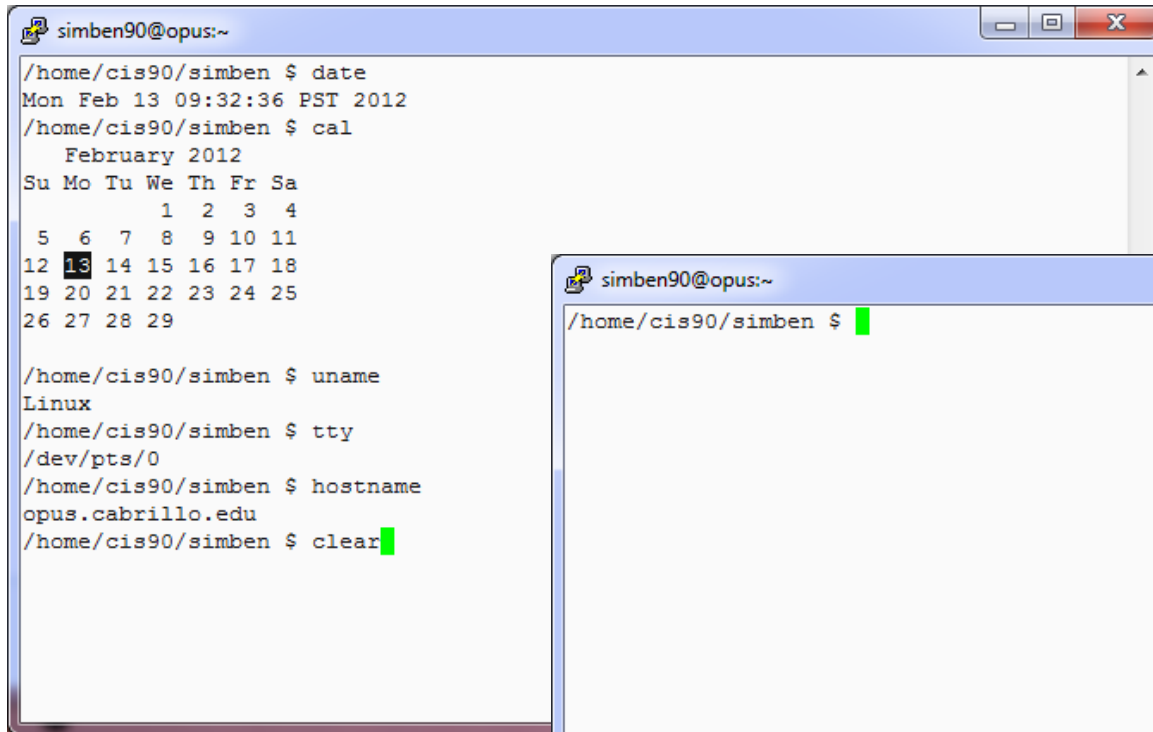
The shell "prompt"

The "command"

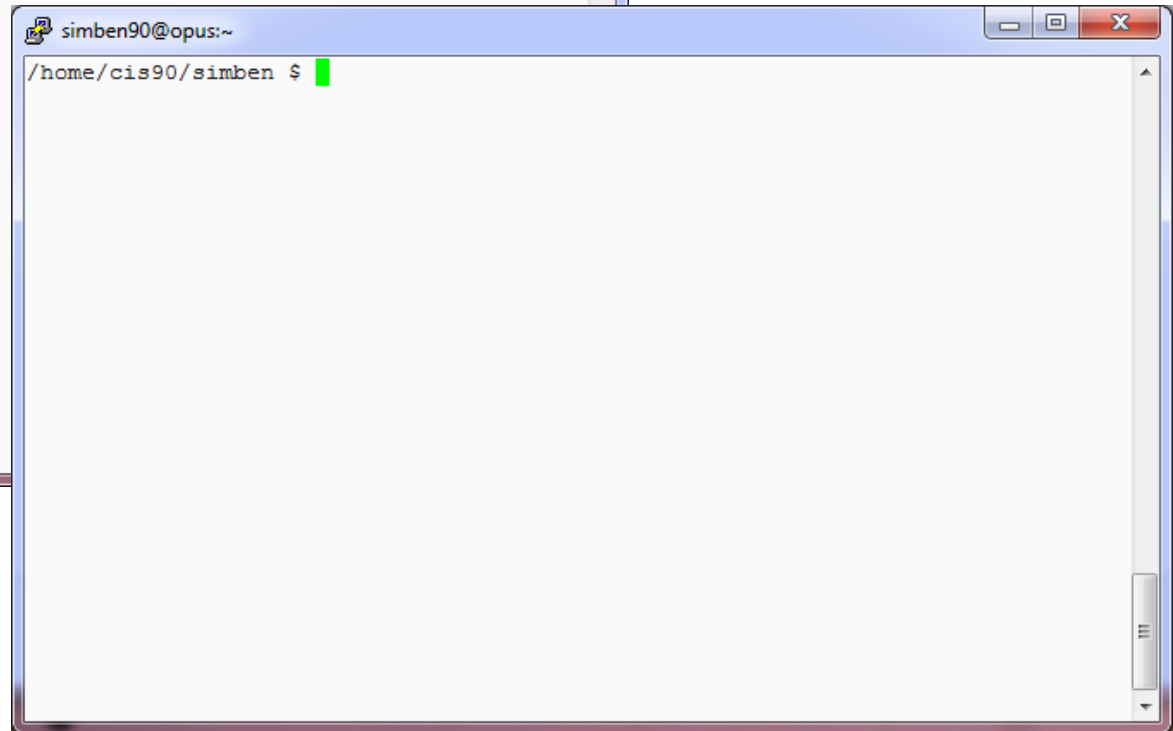
```
/home/cis90/simben $ date  
Sat Sep  1 14:03:33 PDT 2012  
/home/cis90/simben $
```

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

How do I clear the screen?



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

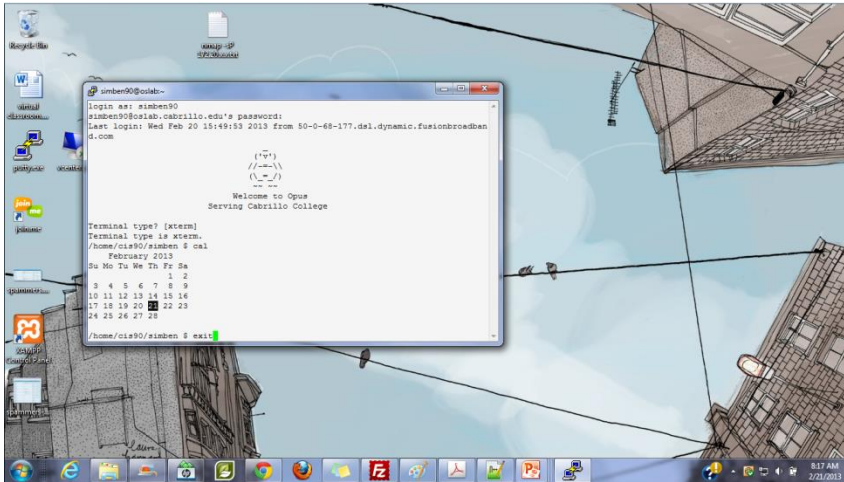


```
simben90@opus:~  
/home/cis90/simben $
```

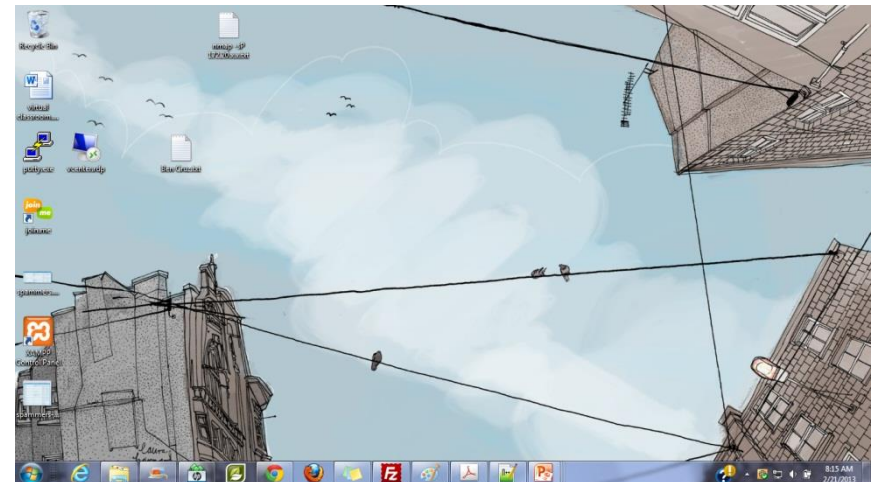
The **clear** command scrolls previous commands out of sight

How do I end this login session?

before **exit**



after **exit**



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

Viewing your command history

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

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

< *snipped* >

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

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

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

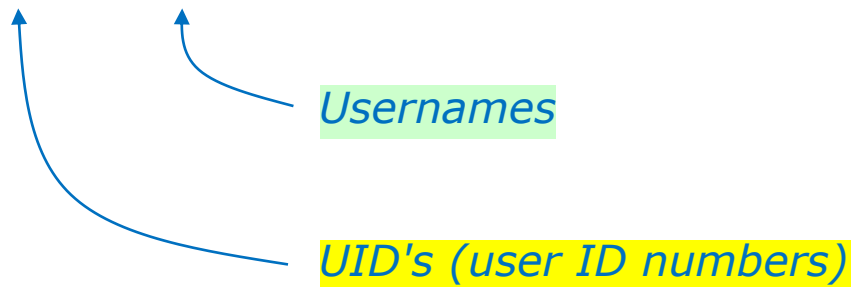
Tip: Use the "Up Arrow" key to quickly re-issue a previous command!

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

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

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

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



We are all just numbers to the Linux kernel

What shell am I using?

Shell	
System Commands	Applications
Kernel	

```

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

Process ID numbers →

Terminal device being used →

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

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

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

How do I log into another computer system?

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

username on remote computer → *Hostname of remote computer*

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

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

Notice how the prompt changes on the remote computer →

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

How do I log into another computer system?

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

username on remote computer

IP address of remote computer

/home/cis90/simben \$ **ssh cis90@172.20.4.34**

cis90@172.20.4.34's password:

Welcome to Ubuntu 12.04.1 LTS (GNU/Linux 3.2.0-29-generic x86_64)

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

361 packages can be updated.

109 updates are security updates.

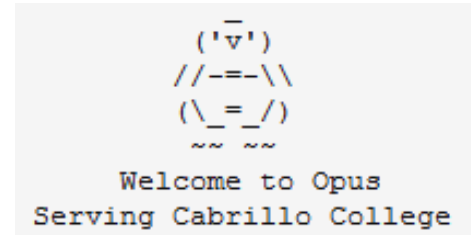
Last login: Wed Feb 20 17:26:25 2013 from oslab.cabrillo.edu

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
```

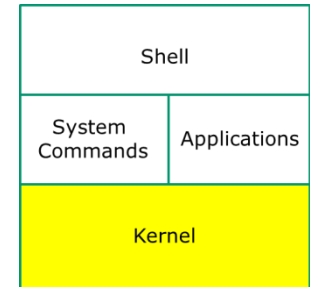


We still refer to Opus as "Opus" in this class however it's official hostname on the Internet is "oslab". This may change in the future after some network changes are made.

Opus is a member of two overlapping Internet domains:

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

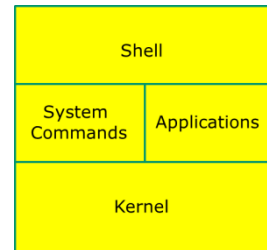
What kernel am I running on?



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

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

What "distro" has been installed?



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

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

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



What terminal device am I using?

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

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

Who else is logged in and from where?

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

username

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

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

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

Which is my login session?

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

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

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

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

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

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

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

Test your knowledge

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

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

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

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

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

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

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

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

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

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

Answer: /dev/pts/0

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

What type of terminal am I using right now?

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

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

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

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

What type of terminal am I using right now?

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

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

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

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

Answer: xterm

We have the answer already!

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

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

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

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

Answer: oslab.cabrillo.edu

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

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

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

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

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

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

Answer: Linux

What is the name of the Linux Distribution being run?

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


What is the name of the Linux Distribution being run?

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

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

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

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

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

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

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

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

Answer: CentOS

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

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

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

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

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

Answer: username=simben90 and the uid=1001

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

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

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

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

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

Answer: bash

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

We will soon learn another command for doing this.